2014

Adaptive discontinuous evolution Galerkin method for dry atmospheric flow

Yelash, L.; Müller, A.; Lukáová-- - Medviová, M.; Giraldo, F.X.; Wirth, V.

AMS

Yelash, L. Müller, A. Lukáová---Medviová, M. Giraldo, F.X., Wirth, V., "Adaptive discontinuous evolution Galerkin method for dry atmospheric flow", Journal of Computational Physics, Volume 268, (1 July 2014), Pages 106-133.
http://hdl.handle.net/10945/55814

Adaptive discontinuous evolution Galerkin method for dry atmospheric flow

L. Yelash[1], A. Müller[2], M. Lukáčová - Medviďová[1], F.X. Giraldo[2] and V. Wirth[3]

March 25, 2014

Abstract

We present a new adaptive genuinely multidimensional method within the framework of the discontinuous Galerkin method. The discontinuous evolution Galerkin (DEG) method couples a discontinuous Galerkin formulation with approximate evolution operators. The latter are constructed using the bicharacteristics of multidimensional hyperbolic systems, such that all of the infinitely many directions of wave propagation are considered explicitly. In order to take into account multiscale phenomena that typically appear in atmospheric flows nonlinear fluxes are split into a linear part governing the acoustic and gravitational waves and a nonlinear part that models advection. Time integration is realized by the IMEX type approximation using the semi-implicit second-order backward differentiation formula (BDF2). Moreover in order to approximate efficiently small scale phenomena, adaptive mesh refinement using the space filling curves via the AMATOS function library is employed. Four standard meteorological test cases are used to validate the new discontinuous evolution Galerkin method for dry atmospheric convection. Comparisons with the Rusanov flux, a standard one-dimensional approximate Riemann solver used for the flux integration, demonstrate better stability and accuracy, as well as the reliability of the new multidimensional DEG method.

Key words: dry atmospheric convection, steady states, systems of hyperbolic balance laws, Euler equations, large time step, semi-implicit approximation, evolution Galerkin schemes

AMS Subject Classification: 65L05, 65M06, 35L45, 35L65, 65M25, 65M15

[1]Institute of Mathematics, University of Mainz, Staudingerweg 9, 55099 Mainz, Germany, emails: {lukacova,yelash}@uni-mainz.de

[2]Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943-5216, USA, email: {fxgirld@nps.edu, amueller@anmr.de}

[3]Institute for Atmospheric Physics, University of Mainz, Becherweg 21, D-55127 Mainz, Germany, email: vwirth@uni-mainz.de

1 Introduction and Meteorological Motivation

A characteristic property of atmospheric flows is their multiscale nature with wave speeds differing by orders of magnitude. If the Mach and Froude numbers are small, the acoustic and gravitational waves are much faster than advection, but only the latter is of primary interest for numerical weather prediction. Naive explicit time integration would yield prohibitively expensive numerical simulations, which makes a suitable splitting of fast and slow waves highly desirable. This idea is not new and has been used extensively in previous meteorological simulations. Many operational nonhydrostatic weather models use split-explicit methods, where different time steps are used for slow and fast waves, respectively, cf. [16], [26], the National Center for Atmospheric Research [25], Pennsylvania State University/National Center for Atmospheric Research [49] and the German Weather Service [41]. Another common approach is based on semi-implicit time discretization; here the fast waves that are of less interest are approximated implicitly, whereas slow advection is treated explicitly. Several methods following this idea can be found, e.g., in [2, 10, 16, 23, 24, 35, 34, 39, 47] to name just a few.

Another characteristic of many atmospheric flows is their multidimensional character with different localized structural phenomena such as, e.g., the cloud-environment interface. A convenient tool to approximate these local structures efficiently is mesh adaptivity. Indeed, adaptive mesh refinement has been applied in the atmospheric sciences quite successfully over the past decades, see, e.g. [42, 3, 6]. Of course, the strategy where and how the mesh has to be refined is a difficult scientific problem and depends on the particular application. The final application we have in mind is the simulation of an evolving cumulus cloud and its interaction with the environment. This is an important meteorological problem, since the evaporative cooling at the cloud-environment boundary and its impact on the cloud evolution are not well understood [34], [21]. Consequently, efficient adaptive numerical schemes can be expected to improve the insight into the underlying physical processes by explicitly resolving the interplay between the larger scales of the cloud environment and the smaller scales inside the cloud and at its boundary. In order to approximate localized structures efficiently, we will work with adaptive meshes using the space filling curves via the AMATOS function library, cf. [5].

In this paper we develop a new semi-implicit genuinely multidimensional method within the framework of the discontinuous Galerkin method. The method is implemented in the discontinuous Galerkin solver by Giraldo and Warburton [17], see also recent results [33, 34] for applications to the Euler equations. However, instead of the Rusanov flux (a standard one-dimensional approximate Riemann solver), that has been used in [33, 34], the flux integration within the discontinuous Galerkin method is now realized by means of a genuinely multidimensional evolution operator. The latter is constructed using the theory of bicharacteristics in order to take all infinitely many directions of wave propagation into account. The approximate evolution operator can be interpreted as a multidimensional numerical flux function. In the finite volume framework the finite volume evolution Galerkin (FVEG) method has been used successfully for various physical applications, e.g., wave propagation in heterogeneous media [1], the Euler equations of gas dynamics [7, 30] and the shallow water equations [12, 22, 27]. In [32] we derived exact integral

representation and approximate evolution operators for three-dimensional hyperbolic conservation laws and presented the results of the FVEG method for the acoustic equation. The FVEG method has been shown to be more accurate than standard FV methods based on the one-dimensional Riemann solver, see also, [28] for further references. In order to illustrate high accuracy, stability and robustness also for the new DEG method we will concentrate on two-dimensional dry atmospheric flows and standard meteorological test cases. Generalization to fully three-dimensional meteorological test cases is the subject of our future study. To improve efficiency of the new DEG scheme on adaptive triangular grids we have also studied its performance when the multidimensional evolution operator is ported to GPUs. In our recent paper [7] we have shown that we can achieve a considerable speedup of 30 (in comparison to a single CPU core) for the calculation of the evolution Galerkin operator for standard meteorological tests cases.

The remainder of the paper is organized as follows. In the next section we describe the mathematical model governing dry atmospheric flow, in Section 3 we derive the discontinuous Galerkin method for spatial discretization and the IMEX type approximation for time discretization. An emphasis is put on a non-standard discretization of the cell interface fluxes by means of multidimensional EG operators, cf. Subsection 3.2. In Section 4 we present the numerical experiments and illustrate the high accuracy and stability of the new EG method and show comparisons with the DG method that uses the standard Rusanov numerical flux.

2 Mathematical Model

We start with the description of the mathematical model. Motion of compressible flows is governed by the Euler equations

$$\begin{aligned} \partial_t \rho + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p\,\mathrm{Id}) &= -\rho g \mathbf{k} \\ \partial_t (\rho\theta) + \nabla \cdot (\rho\theta \mathbf{u}) &= 0\ , \end{aligned} \tag{2.1}$$

where ρ denotes the density, $\mathbf{u}$ velocity, p pressure and θ the potential temperature. Further, g represents the gravitational constant, Id is the identity matrix and $\mathbf{k}$ the unit vector in the vertical direction. Denoting T temperature, the potential temperature can be obtained from the equation of adiabatic process in an ideal gas

$$\theta = T \left(\frac{p_0}{p} \right)^{R/c_p}, \ R = c_p - c_v.$$

We use potential temperature as a variable since it is better suited for generalization to moist atmospheric flow. In order to close the system we determine pressure from the state equation

$$p = p_0 \left(\frac{R\rho\theta}{p_0} \right)^{\gamma},$$

where $\gamma = c_p/c_v$ is the adiabatic constant and $p_0 = 10^5 Pa$ the reference pressure.

Many geophysical flows can be considered as a perturbation of some reference equilibrium state. For example, atmospheric flows are typically represented as a perturbation over the background hydrostatic state $(\bar{\rho}, \bar{\mathbf{u}}(=0), \bar{p}, \bar{\theta})$, cf., e.g. [16], [34],

$$\frac{\partial \bar{p}}{\partial y} = -\bar{\rho} g.$$

Here we assume that $\bar{\theta} = \bar{\theta}(y)$ or a particular case $\bar{\theta} = 300\text{K}$. Then $\bar{\rho} = \frac{p_0}{R\bar{\theta}} \bar{\pi}^{\frac{c_v}{R}}, \bar{p} = p(\bar{\rho}, \bar{\theta}) = p_0 \left(\frac{R\bar{\rho}\bar{\theta}}{p_0} \right)^{\gamma}$ with the Exner pressure $\bar{\pi}(y) := 1 - gy/c_p\bar{\theta}$.

In order to avoid numerical instabilities due to the multiscale behaviour of (2.1) numerical simulations are typically realized for perturbations $\rho' = \rho - \bar{\rho}$, $\theta' = \theta - \bar{\theta}$, $p' = p - \bar{p}$. The latter satisfy the following equation

$$\begin{aligned} \partial_t \rho' + \nabla \cdot (\rho \mathbf{u}) &= 0 \\ \partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u} + p' \,\mathrm{Id}) &= -\rho' g \mathbf{k} \\ \partial_t (\rho\theta)' + \nabla \cdot (\rho\theta \mathbf{u}) &= 0. \end{aligned} \tag{2.2}$$

Our aim in what follows is to approximate (2.2) with the discontinuous Galerkin method. However, instead of the classical one-dimensional numerical flux function we will apply a genuinely multidimensional evolution operator. To this end let us rewrite (2.2) in the form of a hyperbolic balance law

$$\frac{\partial \mathbf{q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{S}(\mathbf{q}), \tag{2.3}$$

where

$$\mathbf{q} = \begin{pmatrix} \rho' \\ \rho \mathbf{u} \\ (\rho\theta)' \end{pmatrix}$$

and

$$\mathbf{F}(\mathbf{q}) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \otimes \mathbf{u} + p' \,\mathrm{Id} \\ \rho\theta \mathbf{u} \end{pmatrix}, \qquad \mathbf{S}(\mathbf{q}) = \begin{pmatrix} 0 \\ -\rho' g \mathbf{k} \\ 0 \end{pmatrix}$$

is the nonlinear flux function and the source term, respectively. We should note that in our numerical experiments we will also use a stabilization via artificial viscosity [46], [34], which results in the following source term

$$\mathbf{S}(\mathbf{q}) = \begin{pmatrix} 0 \\ -\rho' g \mathbf{k} + \nabla \cdot (\mu \rho \nabla \mathbf{u}) \\ \nabla \cdot (\mu \rho \nabla \theta') \end{pmatrix},$$

where $\mu > 0$ is an artificial viscosity parameter that will be specified later.

3 Discontinuous Galerkin method and the multidimensional EG operator

In the last decades the discontinuous Galerkin (DG) method has been used extensively for approximation of partial differential equations, see, e.g., [9, 11, 13, 20, 36, 40] and the references therein. The method is popular due to its flexibility: it is based on the Galerkin, i.e. variational, formulation, allows handling unstructured triangulations of complex geometries, mesh adaptation, and flexible choice of the polynomial basis. Consequently, both mesh ($h-$) and polynomial ($p-$) adaptivity can be applied quite naturally.

In this section we follow [39], [33], [34] and derive the strong formulation of (2.3). Let us divide the computational domain Ω into a finite number of mesh cells Ω_e with a boundary $\partial\Omega_e$. In our numerical experiments we work with triangular elements Ω_e and use the nodal basis functions $\{\psi_j, j = 1, \dots, N\}$, N is the number of degrees of freedom. Now, multiplying (2.3) with a nodal basis $\psi_i(\mathbf{x})$, integrating over Ω_e and applying twice integration by parts we obtain the strong formulation

$$\int_{\Omega_e} \left(\frac{\partial \mathbf{q}_h}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{q}_h) - \mathbf{S}(\mathbf{q}_h) \right) \psi_i(\mathbf{x}) \mathrm{d}\mathbf{x} \quad = \quad \int_{\partial\Omega_e} \left(\mathbf{F}(\mathbf{q}_h) - \mathbf{F}^* \right) \psi_i(\mathbf{x}) \mathrm{d}S, \quad i = 1, \dots, N. \tag{3.1}$$

Here $\mathbf{q}_h$ denotes a numerical solution $\mathbf{q}_h(\mathbf{x}) := \sum_j^N \mathbf{q}_j \psi_j(\mathbf{x})$ and $\mathbf{F}^*$ is a suitable numerical flux function that approximates the cell interface fluxes.

We use the Lagrange polynomials having the basis functions ψ_j with the Fekete points for the interpolation and the Gauss points for the integrations. In the numerical experiments presented below approximations with second order Lagrange polynomials will be used. In the previous papers [15, 17, 33, 34], where geophysical flows were simulated by the discontinuous Galerkin method the one-dimensional Rusanov flux was chosen as the numerical flux, i.e.

$$\mathbf{F}^* := \frac{1}{2} \left[\mathbf{F}(\mathbf{q}_L) + \mathbf{F}(\mathbf{q}_R) - \lambda \left(\mathbf{q}_R - \mathbf{q}_L \right) \mathbf{n} \right]. \tag{3.2}$$

Here $\mathbf{q}_L$, $\mathbf{q}_R$ denote the limiting left and right values of the approximate solution at the cell interface, $\mathbf{n}$ is the outer normal to the cell interface and λ the maximum wave speed $\|\mathbf{u}\|_{L_2} + a$, where a denotes the speed of sound.

The novelty of our work relies on the use of the multidimensional evolution operator in order to compute $\mathbf{F}^*$. More precisely, in the predictor step the cell interface values $\mathbf{q}^*$ are computed by the multidimensional evolution Galerkin operator (EG), $\mathbf{q}^* := EG\mathbf{q}_h$. The multidimensional numerical flux is then obtained as follows

$$\mathbf{F}^* := \mathbf{F}(\mathbf{q}^*). \tag{3.3}$$

Derivation of the multidimensional evolution operator will be presented in Section 3.2. The second order derivatives produced by the diffusion terms are discretized by using the local discontinuous Galerkin method of Bassi and Rebay [4], more details can be found also in [34], Section 3.4.

3.1 IMEX time integration

In order to approximate efficiently multiscale phenomena we follow the ideas of Restelli and Giraldo [14], [39] and split the system (2.3) into two parts governing fast and slow waves. The fast waves are approximated implicitly and the slow waves explicitly. More precisely,

$$\frac{\partial \mathbf{q}}{\partial t} = \mathcal{N}(\mathbf{q}),$$

where the full nonlinear operator $\mathcal{N}(\mathbf{q}) = -\nabla \cdot \mathbf{F}(\mathbf{q}) + \mathbf{S}(\mathbf{q})$ is split in the following way

$$\mathcal{N}(\mathbf{q}) = \mathcal{L}(\mathbf{q}) + \mathcal{R}(\mathbf{q})$$

with $\mathcal{R} := \mathcal{N} - \mathcal{L}$ and

$$\mathcal{L}(\boldsymbol{q}) := -\begin{pmatrix} \operatorname{div}(\rho \boldsymbol{u}) \\ \partial p'/\partial x \\ \partial p'/\partial y + g\rho' \\ \operatorname{div}(\overline{\theta}\rho \boldsymbol{u}) \end{pmatrix}.$$

Here we use a linearized version of gradients of p' and set $\frac{\partial p'}{\partial x} = \frac{c_p \overline{p}}{c_v \overline{\rho}\overline{\theta}} \frac{\partial (\rho\theta)'}{\partial x} = \tilde{\gamma} \frac{\partial (\rho\theta)'}{\partial x}$, where $\tilde{\gamma} = \gamma R = \text{const}$. Consequently, the operator $\mathcal{L}$ is indeed a linear operator with respect to the variables $\mathbf{q} := (\rho', \rho u, \rho v, (\rho\theta)')^T$. Denoting $S(\boldsymbol{q}) := -(0, 0, g\rho', 0)^T$ the semi-discrete form of the linear subsystem reads

$$\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{J}_1 \frac{\partial \boldsymbol{q}}{\partial x} + \boldsymbol{J}_2 \frac{\partial \boldsymbol{q}}{\partial y} = S(\boldsymbol{q}). \tag{3.4}$$

Here the Jacobians are

$$\boldsymbol{J}_1 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \tilde{\gamma} \\ 0 & 0 & 0 & 0 \\ 0 & \bar{\theta} & 0 & 0 \end{pmatrix} \quad \boldsymbol{J}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{\gamma} \\ 0 & 0 & \bar{\theta} & 0 \end{pmatrix},$$

where $\bar{\theta} = \bar{\theta}(y)$.

Now, computing the eigenstructure of the matrix pencil of the system (3.4) shows that (3.4) is a hyperbolic system with eigenvalues $\lambda_1 = -c,\ \lambda_{2,3} = 0, \lambda_4 = c$, where $c := \sqrt{\tilde{\gamma}\overline{\theta}}$. It should be pointed out that for the non-dimensional form of (2.2) we would have $\tilde{\gamma} = \frac{\gamma R}{M^2}$, where $M = u_{ref}/c_{ref}$ is the Mach number and the system (3.4) indeed models fast acoustic waves, see also [38]. Consequently, it has to be approximated in an implicit way. In what follows we use the second order backward difference formula of implicit-explicit (IMEX) type

$$\frac{1}{\gamma \Delta t} \sum_{m=-1}^{1} \alpha_m \boldsymbol{q}_h^{n-m} = \sum_{m=0}^{1} \beta_m \left[\mathcal{N}(\boldsymbol{q}_h^{n-m}) - \mathcal{L}(\boldsymbol{q}_h^{n-m}) \right] + \mathcal{L}(\boldsymbol{q}_h^{n+1}), \tag{3.5}$$

where for the fixed time step Δt we have $\alpha_{-1} = 1, \alpha_0 = 4/3, \alpha_1 = -1/3, \gamma = 2/3, \beta_0 = 2, \beta_1 = -1$.

The scheme can be also rewritten as the explicit predictor and implicit corrector scheme, which yields

$$\boldsymbol{q}_h^{ex} := -\sum_{m=0}^{1} \alpha_m \boldsymbol{q}_h^{n-m} + \gamma \Delta t \sum_{m=0}^{1} \beta_m \mathcal{N}(\boldsymbol{q}_h^{n-m}) \tag{3.6}$$

and

$$[1 - \gamma \Delta t \mathcal{L}]\, \boldsymbol{q}_h^{n+1} = \boldsymbol{q}_h^{ex} - \gamma \Delta t \sum_{m=0}^{1} \beta_m \mathcal{L}(\boldsymbol{q}_h^{n-m}). \tag{3.7}$$

The resulting system of linear algebraic equations is solved by a suitable algebraic solver. In our experiments presented below we have used the GMRES with the Jacobi preconditioner, more sophisticated choices of linear solvers and preconditioners (e.g., see [8]) may further decrease computational costs.

3.2 Multidimensional EG operators

The evolution Galerkin method has been firstly proposed by Lukáčová, Morton and Warnecke [29] for the linear acoustic equation and later generalized by Lukáčová and coworkers in the framework of finite volume evolution Galerkin schemes for more complex hyperbolic conservation laws, such as the Euler equations of gas dynamics [30], the shallow water equations [27, 22, 12] just to mention a few. Since the flux integrals in the finite volume framework are approximated using the multidimensional evolution operators the interaction of complex multidimensional waves is approximated more accurately in comparison to schemes using just one-dimensional approximate Riemann solvers. Extensive numerical experiments also confirm good stability as well as high accuracy of the evolution Galerkin methods [29, 30, 27, 22].

In this subsection we derive approximate evolution operators for both operators $\mathcal{N}$ as well as $\mathcal{L}$ that are based on the theory of bicharacteristics for multidimensional hyperbolic conservation laws. We will describe the derivation of the evolution operator for the operator $\mathcal{N}$ in more details, the derivation of the evolution operator $\mathcal{L}$ is analogous.

First, let us rewrite (2.2) in a quasilinear form using the primitive variables $\mathbf{w} = (\rho', u, v, p')$

$$\partial_t \mathbf{w} + \boldsymbol{A}_1(\mathbf{w})\, \partial_x \mathbf{w} + \boldsymbol{A}_2(\mathbf{w})\, \partial_y \mathbf{w} = \mathbf{s}(\mathbf{w}) \tag{3.8}$$

with

$$\boldsymbol{A}_1 = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & \frac{1}{\rho} \\ 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & u \end{pmatrix}, \quad \boldsymbol{A}_2 = \begin{pmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & \frac{1}{\rho} \\ 0 & 0 & \gamma p & v \end{pmatrix}, \quad \mathbf{s} = -\begin{pmatrix} \partial_y \bar{\rho}\, v \\ 0 \\ \frac{\rho'}{\rho} g \\ \partial_y \bar{p}\, v \end{pmatrix}. \tag{3.9}$$

Using the above thermodynamic relationship for $\bar{\rho}, \bar{p}$ we obtain

$$\partial_y \bar{\rho} = -\frac{p_0\, c_v\, g}{(R\bar{\theta})^2 c_p}\left(1 - \frac{g\bar{y}}{c_p\,\bar{\theta}}\right)^{\frac{c_v}{R}-1} \quad , \quad \partial_y \bar{p} = -\frac{g\, p_0}{R\,\bar{\theta}}\left(1 - \frac{g\bar{y}}{c_p\,\bar{\theta}}\right)^{\frac{c_v}{R}} .$$

We first linearize (3.8) by freezing the Jacobian matrices $\boldsymbol{A}_1, \boldsymbol{A}_2$ at a suitable intermediate state $\tilde{\rho'}$, $\tilde{u}$, $\tilde{v}$, $\tilde{p'}$. In the numerical experiments presented below this intermediate state is obtained as the local average of (left/right) limiting values at cell interfaces. Consequently, the resulting Jacobian matrices are constant matrices $\boldsymbol{A}_1, \boldsymbol{A}_2$ evaluated at $\tilde{\rho'}$, $\tilde{u}$, $\tilde{v}$, $\tilde{p'}$. Since our problem is hyperbolic we have real eigenvalues and a full set of linearly independent eigenvectors corresponding to the linear matrix pencil $\boldsymbol{P} := \boldsymbol{A}_1 n_x + \boldsymbol{A}_2 n_y$, where (n_x, n_y) is an arbitrary unit vector $\|(n_x, n_y)\| = 1$. Indeed, the eigenvalues of $\boldsymbol{P}$ are

$$\begin{aligned}
\lambda_1 &= \tilde{u}\, n_x + \tilde{v}\, n_y - \tilde{a}, \\
\lambda_2 &= \lambda_3 = \tilde{u}\, n_x + \tilde{v}\, n_y, \\
\lambda_4 &= \tilde{u}\, n_x + \tilde{v}\, n_y + \tilde{a},
\end{aligned}$$

where $\tilde{a} := \sqrt{\gamma \frac{\tilde{p}}{\tilde{\rho}}} = \sqrt{\gamma R \tilde{\theta}\left(\frac{\tilde{\rho} R \tilde{\theta}}{p_0}\right)^{\frac{R}{c_v}}}$ is the speed of sound. Now multiplying (3.8) by a matrix $\boldsymbol{R}^{-1}$, $\boldsymbol{R}$ consists of the right eigenvectors of $\boldsymbol{P}$, we can rewrite (3.8) using the so-called characteristic variables $\mathbf{v} = \boldsymbol{R}^{-1}\mathbf{w}$

$$\partial_t \mathbf{v} + \boldsymbol{B}_1 \partial_x \mathbf{v} + \boldsymbol{B}_2 \partial_y \mathbf{v} = \mathbf{r}\,,$$

where $\boldsymbol{B}_i := \boldsymbol{R}^{-1}\boldsymbol{A}_i\boldsymbol{R}$ and $\mathbf{r} := \boldsymbol{R}^{-1}\mathbf{s}(\mathbf{w})$. Equivalently, we have

$$\partial_t \mathbf{v} + \mathrm{diag}(\boldsymbol{B}_1)\partial_x \mathbf{v} + \mathrm{diag}(\boldsymbol{B}_2)\partial_y \mathbf{v} = \mathbf{S} + \mathbf{r} \tag{3.10}$$

with

$$\mathbf{S} := -\left[(\boldsymbol{B}_1 - \mathrm{diag}(\boldsymbol{B}_1))\partial_x \mathbf{v} + (\boldsymbol{B}_2 - \mathrm{diag}(\boldsymbol{B}_2))\partial_y \mathbf{v}\right].$$

Integrating each equation of (3.10) along the corresponding bicharacteristic $\mathbf{x}_j$ from t_n to $t_n + \tau$:

$$\frac{d\mathbf{x}_j}{dt} := [\boldsymbol{B}_{1,jj}, \boldsymbol{B}_{2,jj}]^{\mathsf{T}},\ j = 1, \ldots, 4,$$

we obtain after lengthy manipulations, presented in Appendix, the following exact integral representation

$$\begin{aligned}
\rho'(\mathbf{P}) = \frac{\tilde{\rho}}{2\pi\tilde{a}} \int_0^{2\pi} &\left[-\cos(\omega)\, u(\mathbf{Q}_1(\omega)) - \sin(\omega)\, v(\mathbf{Q}_1(\omega)) + \frac{1}{\tilde{\rho}\,\tilde{a}} p'(\mathbf{Q}_1(\omega))\right] d\omega \\
&+ \rho'(\mathbf{Q}_2) - \frac{p'(\mathbf{Q}_2)}{\tilde{a}^2} \\
&- \frac{\tilde{\rho}}{2\pi\tilde{a}} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \beta(t,\omega)\, dt\, d\omega \\
&- \frac{\tilde{\rho}}{2\pi\tilde{a}} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} -\sin(\omega) g \frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega)) + \frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}} \partial_y \bar{p}\, dt\, d\omega
\end{aligned}$$

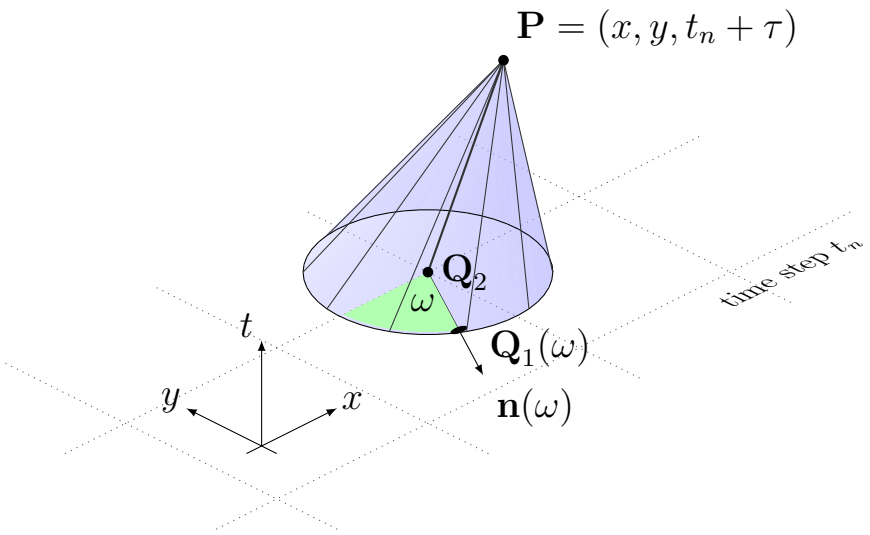

Figure 1: Bicharacteristic cone used for the evolution operator

$$
\begin{aligned}
&+ \int_{t_n}^{t_n+\tau} v(\mathbf{x}_2(t)) \left(-\partial_y \bar{\rho} + \frac{\partial_y \bar{p}}{\tilde{a}^2} \right) dt \,, \\
u(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} &\left[-\frac{p'(\mathbf{Q}_1(\omega))}{\tilde{\rho}\tilde{a}} \cos(\omega) + u(\mathbf{Q}_1(\omega)) \cos^2(\omega) + v(\mathbf{Q}_1(\omega)) \sin(\omega) \cos(\omega) \right] d\omega \\
&+ \frac{1}{2} u(\mathbf{Q}_2) + \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \cos(\omega)\, \beta(t,\omega)\, dt\, d\omega \\
&+ \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \cos(\omega) \frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}} \partial_y \bar{p} - \sin(\omega)\cos(\omega) g \frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega))\, dt\, d\omega \\
&- \frac{1}{2\tilde{\rho}} \int_{t_n}^{t_n+\tau} \partial_x p'(\mathbf{x}_2(t))\, dt \,, \\
v(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} &\left[-\frac{p'(\mathbf{Q}_1(\omega))}{\tilde{\rho}\tilde{a}} \sin(\omega) + u(\mathbf{Q}_1(\omega)) \cos(\omega) \sin(\omega) + v(\mathbf{Q}_1(\omega)) \sin^2(\omega) \right] d\omega \\
&+ \frac{1}{2} v(\mathbf{Q}_2) + \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \sin(\omega)\, \beta(t,\omega)\, dt\, d\omega \\
&+ \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \sin(\omega) \frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}} \partial_y \bar{p} - \sin^2(\omega) g \frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega))\, dt\, d\omega \\
&- \frac{1}{2\tilde{\rho}} \int_{t_n}^{t_n+\tau} \partial_y p'(\mathbf{x}_2(t))\, dt - \frac{1}{2} g \int_{t_n}^{t_n+\tau} \frac{\rho'}{\rho}(\mathbf{x}_2(t))\, dt \,, \\
p'(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} &\left[p'(\mathbf{Q}_1(\omega)) - \tilde{\rho}\tilde{a} u(\mathbf{Q}_1(\omega)) \cos(\omega) - \tilde{\rho}\tilde{a} v(\mathbf{Q}_1(\omega)) \sin(\omega) \right] d\omega \\
&- \tilde{\rho}\tilde{a} \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \beta(t,\omega)\, dt\, d\omega
\end{aligned}
$$

$$-\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau} -\sin(\omega)\tilde{\rho}\tilde{a}\, g\frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega)) + v(\mathbf{x}_1(t,\omega))\partial_y\bar{p}\, dt\, d\omega. \quad (3.11)$$

Here $\beta(t,\omega) = \tilde{a}\,[\partial_x u\, \sin^2(\omega) - (\partial_y u + \partial_x v)\sin(\omega)\cos(\omega) + \partial_y v\, \cos^2(\omega)]$ and $\mathbf{P} = (x, y, t_n + \tau)$, $\mathbf{Q}_1 \equiv \mathbf{Q}_1(\omega) = (x - (\tilde{u} - \tilde{a}\cos(\omega))\tau,\, y - (\tilde{v} - \tilde{a}\sin(\omega))\tau,\, t_n)$, $\mathbf{Q}_2 = (x - \tilde{u}\tau,\, y - \tilde{v}\tau,\, t_n)$ are respectively the peak and foot points of the bicharacteristics that generate the mantle of the bicharacteristic cone, cf. Figure 1.

To obtain an explicit approximate evolution operator the above exact integral representation needs to be approximated. First, all time integrals are approximated using the rectangle rule. Integrals along the base perimeter, that contain $\beta(t_n,\omega)$ terms, are replaced by means of integration by parts. More precisely, applying Lemma 2.1 [29] we obtain

$$\tau\int_0^{2\pi}\beta(t_n,\omega) = \int_0^{2\pi} u(\mathbf{Q}_1)\cos(\omega) + v(\mathbf{Q}_1)\sin(\omega)\, d\omega$$

$$\begin{aligned}
\tau\int_0^{2\pi}\beta(t_n,\omega)\cos(\omega) &= \int_0^{2\pi} u(\mathbf{Q}_1)(2\cos^2(\omega)-1) + 2v(\mathbf{Q}_1)\sin(\omega)\cos(\omega)\, d\omega \\
\tau\int_0^{2\pi}\beta(t_n,\omega)\sin(\omega) &= \int_0^{2\pi} 2u(\mathbf{Q}_1)\sin(\omega)\cos(\omega) + v(\mathbf{Q}_1)(2\sin^2(\omega)-1)\, d\omega.
\end{aligned}$$

Further we approximate $\frac{\rho'}{\rho}$ with $\frac{\rho'}{\tilde{\rho}}$ and substitute the condition for hydrostatic balance $\partial_y\bar{p} = -\bar{\rho}g$ in the equations for $\rho'(\mathbf{P}), u(\mathbf{P}), v(\mathbf{P}), p'(\mathbf{P})$. The state equation also yields

$$\partial_y\bar{\rho} = -\pi^{-1}\frac{c_v}{c_p\, R\bar{\theta}}\bar{\rho}g = -\frac{\bar{\rho}g}{\bar{a}^2}.$$

This implies that the last term in the equations for $\rho'(\mathbf{P})$ approximately vanishes

$$\int_{t_n}^{t_n+\tau} v(\mathbf{x}_2(t))\left(-\partial_y\bar{\rho} + \frac{\partial_y\bar{p}}{\tilde{a}^2}\right) dt \approx 0.$$

Moreover, pressure derivatives $\partial_x p'(\mathbf{x}_2)$ and $\partial_y p'(\mathbf{x}_2)$ in the equations for $u(\mathbf{P}), v(\mathbf{P})$ can be approximated using the midpoint rule and the Gauss theorem

$$\begin{aligned}
-\frac{\tau}{2\tilde{\rho}}\partial_x p'(\mathbf{Q}_2) &= -\frac{1}{2\pi\tilde{\rho}\tilde{a}}\int_0^{2\pi} p'(\mathbf{Q}_1)\cos(\omega)\, d\omega + \mathcal{O}(\tau^2) \\
-\frac{\tau}{2\tilde{\rho}}\partial_y p'(\mathbf{Q}_2) &= -\frac{1}{2\pi\tilde{\rho}\tilde{a}}\int_0^{2\pi} p'(\mathbf{Q}_1)\sin(\omega)\, d\omega + \mathcal{O}(\tau^2).
\end{aligned}$$

These approximations yield finally the desired approximate evolution operator in order to predict the cell interface values $\mathbf{q}^* \equiv (\rho'(\mathbf{P}), u(\mathbf{P}), v(\mathbf{P}), p'(\mathbf{P})) = EG^{\mathcal{N}}\,\mathbf{q}_h$ in (3.1).

Approximate evolution operator $EG^{\mathcal{N}}$ for the nonlinear operator $\mathcal{N}$ (3.8)

$$\rho'(\mathbf{P}) = \frac{1}{2\pi\,\tilde{a}} \int_0^{2\pi} \left[-2\tilde{\rho}\cos(\omega)\,u(\mathbf{Q}_1(\omega)) - 2\tilde{\rho}\sin(\omega)\,v(\mathbf{Q}_1(\omega)) + \frac{p'(\mathbf{Q}_1(\omega))}{\tilde{a}} \right] d\omega$$
$$+ \rho'(\mathbf{Q}_2) - \frac{p'(\mathbf{Q}_2)}{\tilde{a}^2} + \frac{\tau}{2\pi\,\tilde{a}} \int_0^{2\pi} \left[\sin(\omega)\rho'(\mathbf{Q}_1(\omega))\,g + \frac{v(\mathbf{Q}_1(\omega))}{\tilde{a}}\bar{\rho}g \right] d\omega\,,$$

$$u(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} \left[-\frac{2p'(\mathbf{Q}_1(\omega))}{\tilde{\rho}\tilde{a}}\cos(\omega) + u(\mathbf{Q}_1(\omega))(3\cos^2(\omega) - 1) + 3v(\mathbf{Q}_1(\omega))\sin(\omega)\cos(\omega) \right] d\omega$$
$$+ \frac{1}{2}u(\mathbf{Q}_2) - \frac{\tau}{2\pi\tilde{\rho}} \int_0^{2\pi} \cos(\omega) \left[\sin(\omega)\rho'(\mathbf{Q}_1(\omega))g + \frac{v(\mathbf{Q}_1(\omega))}{\tilde{a}}\bar{\rho}g \right] d\omega\,,$$

$$v(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} \left[-\frac{2p'(\mathbf{Q}_1(\omega))}{\tilde{\rho}\tilde{a}}\sin(\omega) + 3u(\mathbf{Q}_1(\omega))\sin(\omega)\cos(\omega) + (3\sin^2(\omega) - 1)v(\mathbf{Q}_1(\omega)) \right] d\omega$$
$$+ \frac{1}{2}v(\mathbf{Q}_2) - \frac{\tau}{2\pi\tilde{\rho}} \int_0^{2\pi} \sin(\omega) \left[\sin(\omega)\rho'(\mathbf{Q}_1(\omega))\,g + \frac{v(\mathbf{Q}_1(\omega))}{\tilde{a}}\bar{\rho}g \right] d\omega - \frac{\tau}{2\tilde{\rho}}\rho'(\mathbf{Q}_2)g\,,$$

$$p'(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} \left[p'(\mathbf{Q}_1(\omega)) - 2\tilde{\rho}\tilde{a}u(\mathbf{Q}_1(\omega))\cos(\omega) - 2\tilde{\rho}\tilde{a}v(\mathbf{Q}_1(\omega))\sin(\omega) \right] d\omega$$
$$+ \frac{\tau\,\tilde{a}}{2\pi} \int_0^{2\pi} \left[\sin(\omega)\rho'(\mathbf{Q}_1(\omega))\,g + \frac{v(\mathbf{Q}_1(\omega))}{\tilde{a}}\bar{\rho}g \right] d\omega. \tag{3.12}$$

We should point out that all integrals along the base perimeter (sonic circle), i.e. integrals with respect to ω, are evaluated exactly for given discrete data. We make a transformation of the actual triangle to the reference triangle, where the corresponding integrals along the arcs of the sonic circles were precomputed with the help of the computer algebra package Mathematica.

Our next goal is to derive the approximate evolution operator for the system (3.4). Since the acoustic subsystem (3.4) is linear, the derivation of the exact evolution operator is even easier and it has been already obtained, e.g., in [30]. Indeed, applying the above procedure we obtain in an analogous way the following exact evolution operator for the acoustic system (3.4), cf. [30], [31].

$$\rho'(\mathbf{P}) = \rho'(\mathbf{Q}_2) - \frac{\tilde{\gamma}(\rho\theta)'(\mathbf{Q}_2)}{\tilde{c}^2}$$
$$+ \frac{1}{2\pi\tilde{c}} \int_0^{2\pi} -\cos(\omega)\,(\rho u)(\mathbf{Q}_1(\omega) - \sin(\omega)\,(\rho v)(\mathbf{Q}_1(\omega)) + \frac{\tilde{\gamma}}{\tilde{c}}(\rho\theta)'(\mathbf{Q}_1(\omega))d\omega$$
$$- \frac{1}{2\pi\tilde{c}} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \frac{1}{t_n+\tau-t} \left[\cos(\omega)(\rho u)(\mathbf{x}_1(t,\omega)) + \sin(\omega)(\rho v)(\mathbf{x}_1(t,\omega))\right] dt\, d\omega$$
$$+ \frac{1}{2\pi\tilde{c}} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \sin(\omega) g\rho'(\mathbf{x}_1(t,\omega))\, dt\, d\omega\,,$$

$$(\rho u)(\mathbf{P}) = \frac{1}{2}(\rho u)(\mathbf{Q}_2) - \frac{\tilde{\gamma}}{2} \int_{t_n}^{t_n+\tau} \partial_x(\rho\theta)'(\mathbf{x}_2(t))dt$$
$$+ \frac{1}{2\pi} \int_0^{2\pi} \cos^2(\omega)\,(\rho u)(\mathbf{Q}_1(\omega)) + \sin(\omega)\cos(\omega)\,(\rho v)(\mathbf{Q}_1(\omega)) - \frac{\tilde{\gamma}}{\tilde{c}}(\rho\theta)'(\mathbf{Q}_1(\omega))\cos(\omega)d\omega$$
$$+ \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \frac{1}{t_n+\tau-t} \left[\cos(2\omega)\,(\rho u)(\mathbf{x}_1(t,\omega)) + \sin(2\omega)\,(\rho v)(\mathbf{x}_1(t,\omega))\right] dt\, d\omega$$
$$- \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \sin(\omega)\cos(\omega) g\rho'(\mathbf{x}_1(t,\omega))\, dt\, d\omega\,,$$

$$(\rho v)(\mathbf{P}) = \frac{1}{2}(\rho v)(\mathbf{Q}_2) - \frac{\tilde{\gamma}}{2} \int_{t_n}^{t_n+\tau} \partial_y(\rho\theta)'(\mathbf{x}_2(t))dt - \frac{g}{2} \int_{t_n}^{t_n+\tau} \rho'(\mathbf{x}_2(t))dt$$
$$+ \frac{1}{2\pi} \int_0^{2\pi} \cos(\omega)\sin(\omega)\,(\rho u)(\mathbf{Q}_1(\omega)) + \sin^2(\omega)\,(\rho v)(\mathbf{Q}_1(\omega)) - \frac{\tilde{\gamma}}{\tilde{c}}(\rho\theta)'(\mathbf{Q}_1(\omega))\sin(\omega)d\omega$$
$$+ \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \frac{1}{t_n+\tau-t} \left[\sin(2\omega)\,(\rho u)(\mathbf{x}_1(t,\omega)) - \cos(2\omega)\,(\rho v)(\mathbf{x}_1(t,\omega))\right] dt\, d\omega$$
$$- \frac{1}{2\pi} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \sin^2(\omega) g\rho'(\mathbf{x}_1(t,\omega))\, dt\, d\omega\,,$$

$$(\rho\theta)'(\mathbf{P}) = \frac{1}{2\pi} \int_0^{2\pi} (\rho\omega)'(\mathbf{Q}_1(\theta)) - \frac{\tilde{c}}{\tilde{\gamma}}\cos(\omega)\,(\rho u)(\mathbf{Q}_1(\omega)) - \frac{\tilde{c}}{\tilde{\gamma}}\sin(\omega)\,(\rho v)(\mathbf{Q}_1(\omega))d\omega$$
$$- \frac{\tilde{c}}{2\pi\tilde{\gamma}} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \frac{1}{(t_n+\tau-t)} \left[\cos(\omega)\,(\rho u)(\mathbf{x}_1(t,\omega)) + \sin(\omega)\,(\rho v)(\mathbf{x}_1(t,\omega))\right] dt\, d\omega$$
$$+ \frac{\tilde{c}}{2\pi\tilde{\gamma}} \int_0^{2\pi} \int_{t_n}^{t_n+\tau} \sin(\omega) g\rho'(\mathbf{x}_1(t,\omega))\, dt\, d\omega. \tag{3.13}$$

Here $\tilde{c} = \sqrt{\tilde{\gamma}\bar{\tilde{\theta}}}$ and $\bar{\tilde{\theta}}$ is obtained as a local average of $\bar{\theta}$. Note that we have used analogous notation for bicharacteristics $\mathbf{x}_1(t,\omega)$, $\mathbf{x}_2(t)$ and $\mathbf{Q}_1$, $\mathbf{Q}_2$ as for the nonlinear operator above. Of course, in the case of the linear operator $\mathcal{L}$ the corresponding eigenvalues

determine the geometry of the bicharacteristic cone. Consequently, we now have $\mathbf{Q}_1 = (x - \tilde{c}\cos(\omega)\tau, y - \tilde{c}\sin(\omega)\tau, t_n)$, $\mathbf{Q}_2 = (x, y, t_n)$ and $\mathbf{P} = (x, y, t_n + \tau)$.

In order to approximate the above exact integral representation we apply the rectangle rule for time integrations and the second and third equation of (3.4) to eliminate space derivatives of $(\rho\theta)'(\mathbf{x}_2(t))$ in the equations for ρu and ρv. This yields the approximate evolution operator $EG^{\mathcal{L}}$ that has been denoted as EG1 in [31]. Consequently, we have for the acoustic system (3.4) $\mathbf{q}^* \equiv (\rho'(\mathbf{P}), \rho u(\mathbf{P}), \rho v(\mathbf{P}), (\rho\theta)'(\mathbf{P})) = EG^{\mathcal{L}}\,\mathbf{q}_h$.

Approximate evolution operator $EG^{\mathcal{L}}$ for the linear operator $\mathcal{L}$ (3.4)

$$\rho'(\mathbf{P}) = \rho'(\mathbf{Q}_2) - \frac{\tilde{\gamma}(\rho\theta)'(\mathbf{Q}_2)}{\tilde{c}^2} + \frac{1}{2\pi\tilde{c}}\int_0^{2\pi}\left[-2\cos(\omega)\,(\rho u)(\mathbf{Q}_1(\omega) - 2\sin(\omega)\,(\rho v)(\mathbf{Q}_1(\omega)) + \frac{\tilde{\gamma}}{\tilde{c}}(\rho\theta)'(\mathbf{Q}_1(\omega))\right]d\omega$$

$$(\rho u)(\mathbf{P}) = \frac{1}{\pi}\int_0^{2\pi}\Big[(3\cos^2(\omega) - 1)\,(\rho u)(\mathbf{Q}_1(\omega)) + 3\sin(\omega)\cos(\omega)\,(\rho v)(\mathbf{Q}_1(\omega)) - \frac{\tilde{\gamma}}{\tilde{c}}(\rho\theta)'(\mathbf{Q}_1(\omega))\cos(\omega)\Big]d\omega$$

$$(\rho v)(\mathbf{P}) = -\frac{\tau}{\pi}\int_0^{2\pi}\sin^2(\omega)g\rho'(\mathbf{Q}_1(\omega))\,d\omega + \frac{1}{\pi}\int_0^{2\pi}\Big[3\cos(\omega)\sin(\omega)\,(\rho u)(\mathbf{Q}_1(\omega)) + (3\sin^2(\omega) - 1)\,(\rho v)(\mathbf{Q}_1(\omega)) - \frac{\tilde{\gamma}}{\tilde{c}}(\rho\theta)'(\mathbf{Q}_1(\omega))\sin(\omega)\Big]d\omega$$

$$(\rho\theta)'(\mathbf{P}) = \frac{1}{2\pi}\int_0^{2\pi}\left[(\rho\theta)'(\mathbf{Q}_1(\theta)) - 2\frac{\tilde{c}}{\tilde{\gamma}}\cos(\omega)\,(\rho u)(\mathbf{Q}_1(\omega)) - 2\frac{\tilde{c}}{\tilde{\gamma}}\sin(\omega)\,(\rho v)(\mathbf{Q}_1(\omega))\right]d\omega \quad . \tag{3.14}$$

The resulting discontinuous evolution Galerkin scheme is based on the spatial discretization (3.1), IMEX type time discretization (3.6), (3.7) and the multidimensional approximate evolution operators (3.12) and (3.14) in order to approximate the cell interface fluxes for the nonlinear and linear operators $\mathcal{N}$ and $\mathcal{L}$, respectively. The parameter τ describes the time step of the evolution. In our numerical experiments we have taken $\tau \ll \Delta t$, where Δt is the time step of the IMEX time integration, cf. Section 3.1. Consequently, the operator models the local evolution. In the numerical experiments presented below piecewise quadratic approximate functions in space are used, that yield six degrees of freedom (DoF) per a triangular finite element: three DoFs are located at the vertices of

the triangular mesh cells and three DoFs at the centers of the cell interfaces. When we benchmark our model comparing it to the DG method with the Rusanov flux function, we do not use any kind of limiter or filter except for the artificial viscosity as described in Section 2. In our recent work [7] we have also generalized the discontinuous Galerkin solver of Giraldo and Warburton [17] by including the GPU implementation of the genuinely multidimensional EG operator.

4 Numerical Experiments

To verify the new discontinuous evolution Galerkin method, we carry out four numerical experiments for the test cases: (i) free convection of a smooth warm air bubble, (ii) collision of a large warm bubble with a small cold bubble placed on top of the warm one (iii) a density current caused by a falling cold air bubble, and (iv) the inertia gravity experiment. In experiments (i)-(iii), due to the buoyancy forces caused by differences in the density of the air bubbles and the neutrally stratified environment, the initially resting air bubbles develop a vertical motion with low Mach number $M \ll 1$. In the last experiment, the initial perturbation is advected horizontally and irradiates waves propagating along a periodic horizontal channel.

4.1 Test 1: Free convection of a smooth warm air bubble

This test case was first presented by Giraldo and Restelli in [15], see also [33], [34]. In [37] a similar test case with the infinitely smooth initial data was proposed by Robert, we refer the reader also to further papers, e.g., [18, 44, 45], where similar thermal convection problems have been studied. Here we confine ourselves to the test case from [15] since it has also been computed in [34] using the same DG scheme as in the present paper but applying the Rusanov numerical flux, which is the method that we use to compare against our newly developed DEG scheme.

In this experiment, cf. Figure 2, the warm bubble rises and deforms symmetrically due to the shear friction with the surrounding air at the warm/cold air interface, forming a mushroom-like shape gradually. The warm air bubble is placed at $x_c = 500\text{m}$, $y_c = 350\text{m}$ with the initial potential temperature perturbation:

$$\theta' = \begin{cases} 0 & \text{for} \quad r > r_c \\ (\theta'_c/2)\left[1 + \cos\left(\pi r / r_c\right)\right] & \text{for} \quad r \leq r_c \end{cases} \tag{4.1}$$

where $\theta'_c = 0.5\,\text{K}$ is the maximal initial amplitude of the perturbation, $r_c = 250\text{m}$ is the bubble radius, and r the distance to the center of the bubble (x_c, y_c).

In order to simulate efficiently localized structures arising in geophysical flows, such as cloud boundaries, adaptive mesh refinement is a very suitable tool. We work with the function library AMATOS of Behrens et al. [5], where $h-$adaptive mesh refinement is based on the space filling curve approach. Our numerical experiments were performed

on the domain of 1km×1km with no-flux boundary conditions, using the $h-$adaptive mesh refinement method, where the spatial resolution is adapted by refining or coarsening the mesh cells. The maximal resolution degree of the mesh is $n = 11$, which yields $1000\text{m}/\sqrt{2}^{n+1} \approx 15.6\text{m}$ per the finite element shortest edge in the simulation domain. Since we use second order polynomials, the finest resolution of the grid is 7.8m corresponding to a half of the length of the shortest edge.

Similarly to [34], in the numerical experiment presented in Figure 2 we use a slightly modified, simple refinement criterion

$$\max_{\mathbf{x}\in\Omega_e} [\text{sgn}(\theta'_{\text{c}})\theta'(\mathbf{x}, t)] \geq \sigma|\theta'_{\text{c}}| \tag{4.2}$$

for the deviation of the potential temperature from the background state $\theta' = \theta - \bar{\theta}$; $\sigma \ll 1$ is a test dependent parameter (for the numerical experiments in this work we use $\sigma = 0.1$), and θ'_{c} is the maximal initial amplitude for the perturbation of the potential temperature. For realistic applications, such as cloud simulations, more advanced refinement criteria are needed. The development of refinement criteria is a topic of ongoing research and we will consider different current available options such as physics based criteria or residuum based error estimators in our future research.

If condition (4.2) holds on some element Ω_e, the element will be recursively refined up to a specified finest mesh resolution. In the rest of the computational domain the mesh is adaptively coarsened, see also [34] for further details.

In Figure 2 we show the simulation results obtained using the newly derived multidimensional DEG method (left-hand side of the shown snapshots) and the DG method with the standard Rusanov numerical flux (3.2) (right-hand side). The latter model will be simply referred to as the Rusanov method later on in this paper. Both simulations were performed using the IMEX type approximation for time discretization as described in Section 3.1. The artificial viscosity in both tests is kept constant $\mu = 0.1$ over the computational domain and time. The integration time step is $\Delta t = 0.1\text{s}$. The corresponding Courant-Friedrichs-Lewy conditions calculated on the smallest mesh element is $CFL \equiv \max\left[\frac{(\|\mathbf{u}\|+a)\Delta t}{\Delta x}\right] \approx 4.43$ for the total velocity and $CFL_u \equiv \max\left[\frac{\|\mathbf{u}\|\Delta t}{\Delta x}\right] \leq 3.3{\cdot}10^{-2}$ for the advective part. The time step for the evolution of EG operator is $\tau = 1.25 \cdot 10^{-3}\text{s}$, which corresponds to $CFL_{\text{EG}} \approx 5.5 \cdot 10^{-2}$.

Both models yield very similar solutions of the Euler equations model for the time $t <$ 600s. The differences between both solutions become however noticeable at $t = 600\text{s}$. The solution obtained with the Rusanov flux exhibits short length scale oscillations (cf. isolines at $x \approx 350\text{m}$, $y \approx 800$ in Figure 2, right-hand side, $t = 600\text{s}$) whereas the isolines are quite smooth for the DEG method (on the left-hand side in the same figure). A very pronounced oscillation of the perturbation front can be observed on the top of the air bubble for the Rusanov flux. The snapshots for the time $t = 900\text{s}$ in the same figure show that the solutions become very different for later time.

In order to analyse in more detail the evolution of the solutions at later times we plot a few selected iso-levels of the excess potential temperature for intermediate times in

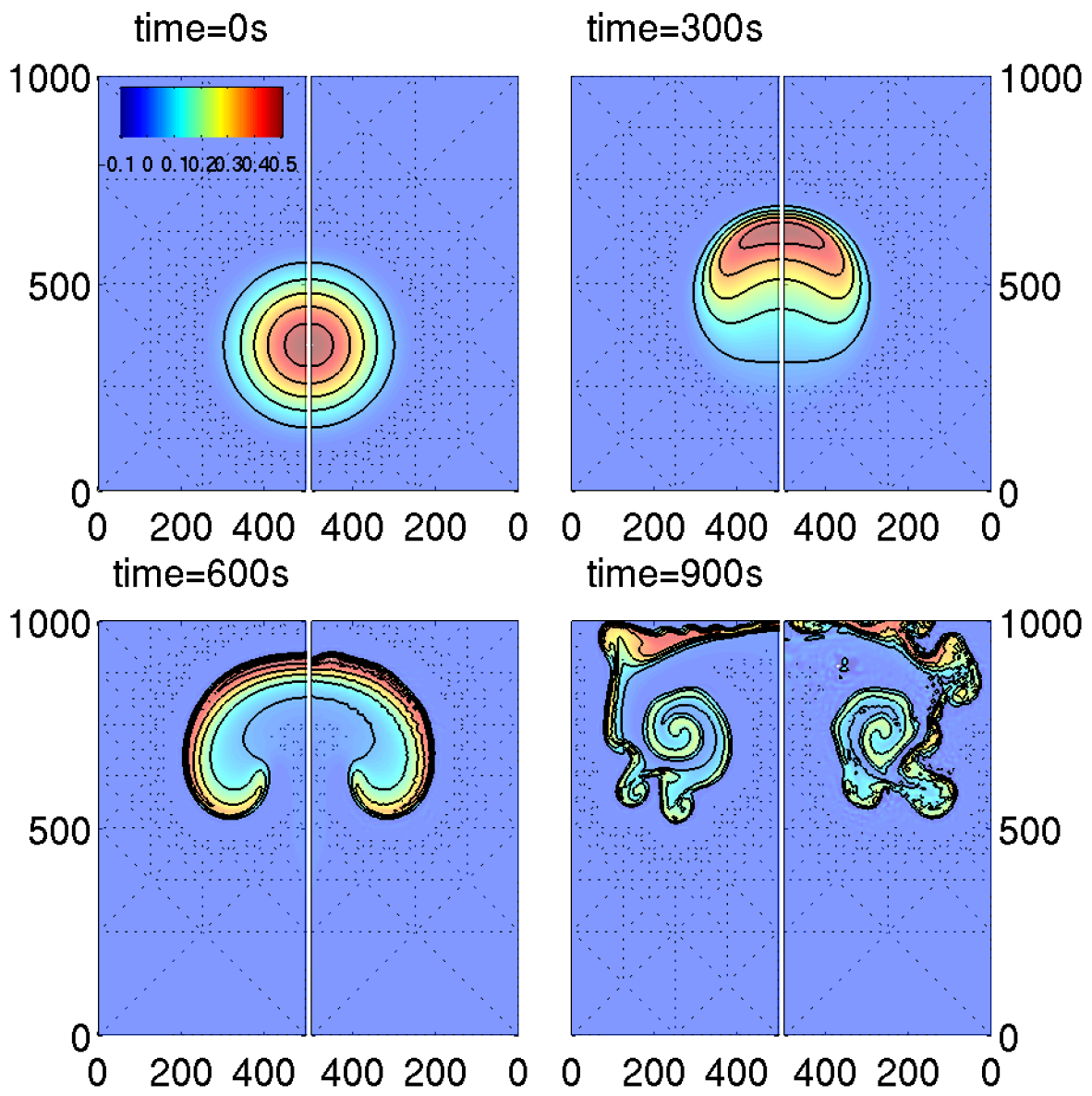

Figure 2: Excess potential temperature θ' for the rising smooth thermal bubble in the numerical experiment proposed by Giraldo and Restelli [15], obtained by the new semi-implicit method on the $h-$adaptive grid with the coarse/fine grid resolution levels $n = 1-11$, respectively, and the constant artificial viscosity $\mu = 0.1\text{m}^2/\text{s}$: On the left-hand side is with the novel multidimensional EG operator in the numerical flux function, on the right-hand side with the Rusanov flux function. The real-world domain is 1km×1km (only a half of the squared computational domain is shown in the x-direction); the shortest edge of the adaptive mesh elements is ≈ 15.6m, the spatial resolution ≈ 7.8m. The simulation times are as indicated; $CFL \approx 4.43$, advective $CFL_{\text{u}} \in \{0, 0.0238, 0.0331, 0.0286\}$ for DEG method and $CFL_{\text{u}} \in \{0, 0.0238, 0.0332, 0.0313\}$ for the Rusanov flux model, $CFL_{\text{EG}} \approx 0.055$. Contour levels correspond to $\theta' \in \{0.05, 0.15, 0.25, 0.35, 0.45\}$K.

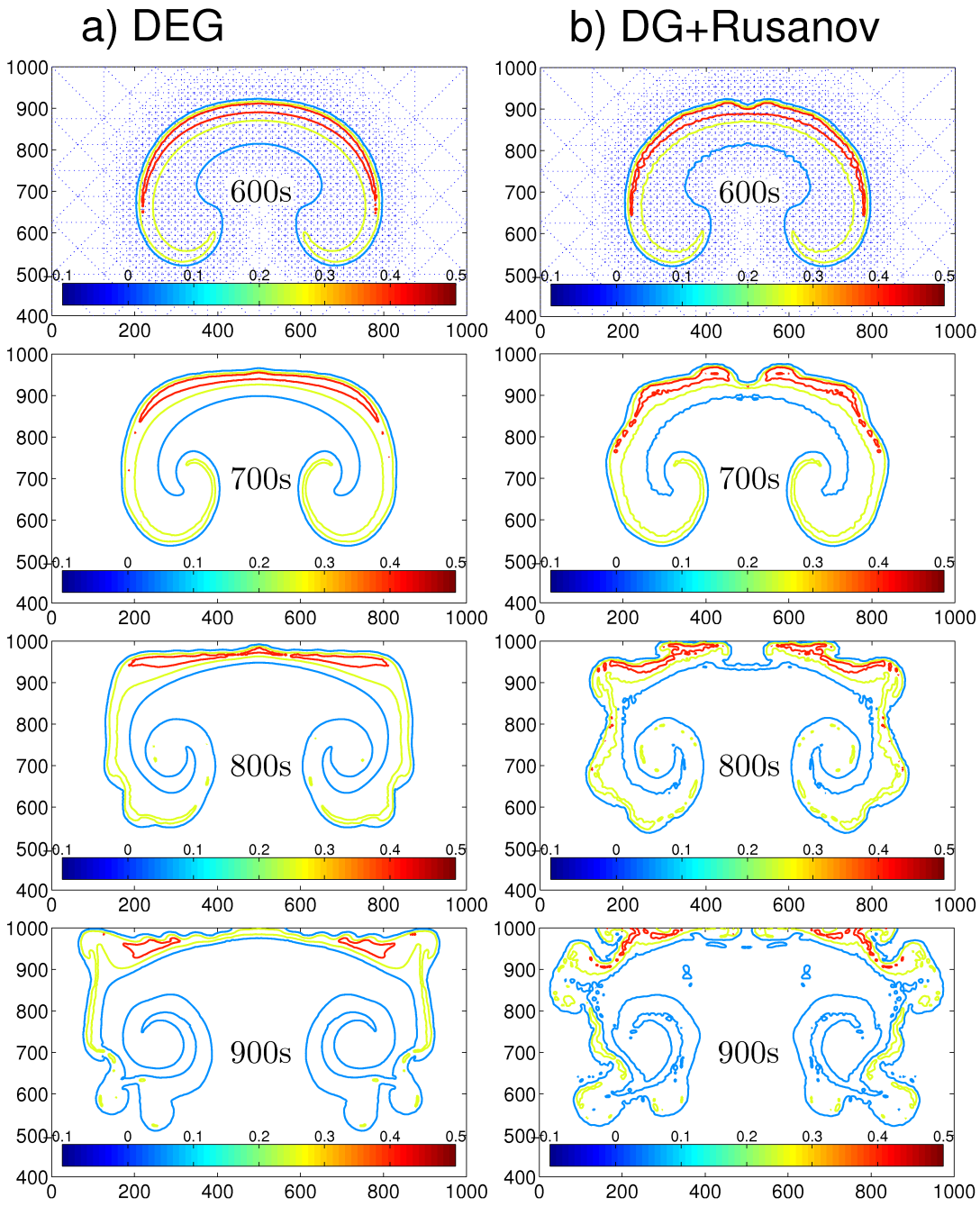

Figure 3: Giraldo-Restelli test for the DEG method a) and the DG method with the Rusanov flux b), with constant viscosity $\mu = 0.1\text{m}^2/\text{s}$ at times as shown; $CFL \approx 4.43$, advective $CFL_{\text{u}} \in \{0.0331, 0.0312, 0.0278, 0.0286\}$ for DEG model (from top to bottom) and $CFL_{\text{u}} \in \{0.0332, 0.0314, 0.0279, 0.0313\}$ for the Rusanov flux. Contour levels are for $\theta' \in \{0.05, 0.25, 0.4\}$K. The structure of the adaptive meshes is shown once for clarity.

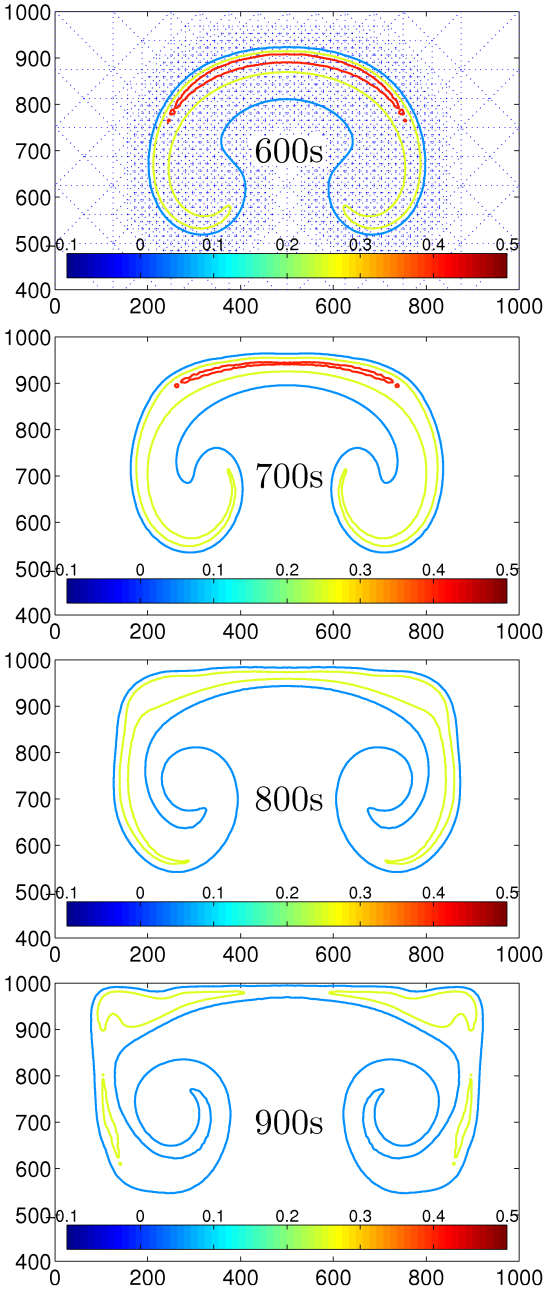

Figure 4: Giraldo-Restelli test for the DG method with the Rusanov flux, with constant viscosity $\mu = 0.25\text{m}^2/\text{s}$ at times as shown, $CFL \approx 4.43$, $CFL_{\text{u}} \in \{0.0306, 0.0299, 0.0269, 0.0248\}$ (from top to bottom). Contour levels are for $\theta' \in \{0.05, 0.25, 0.4\}$K.

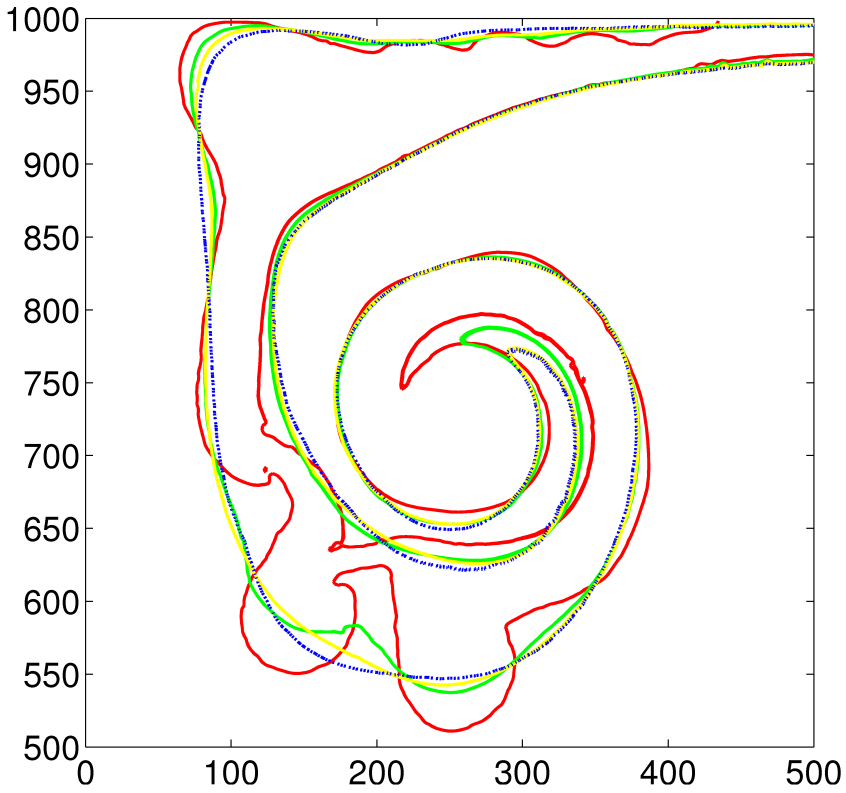

Figure 5: Comparison of the contour level $\theta' = 0.05\,\mathrm{K}$ of the excess potential temperature in the Giraldo and Restelli test for time $t = 900\mathrm{s}$ using the DEG method with constant viscosity $\mu = \{0.1, 0.18, 0.25\}\mathrm{m}^2/\mathrm{s}$ (red, green, and yellow curves, respectively), and the Rusanov flux model with constant viscosity $\mu = 0.25\mathrm{m}^2/\mathrm{s}$ (blue curves). The coarse/fine mesh resolution is $n = 1 - 11$, corresponding to the spatial resolution of $\approx 7.8\mathrm{m}$ along the shortest element edge (15.6m), $CFL \approx 4.43$, $CFL_{\mathrm{u}} \approx 0.0286$ (DEG) and $CFL_{\mathrm{u}} \approx 0.0248$ (Rusanov).

Figure 3. The isolines shown for $\theta' = 0.05$, 0.25, and 0.4 K represent the structure of the solution close to the background, in the middle, and near the maximum of the background perturbation, respectively. The solution based on the EG operator can be characterized as being stable against the short length scale perturbations (all isolines are smooth). A large scale oscillation appears at the bottom of the air bubble interface at time $t = 800$s ($x \approx 850$m, $y \approx 600$m) that seems to be due to the Kelvin-Helmholtz instability caused by the shear flow interactions along the interface between the rising bubble with the surrounding air at rest. These long wave-length oscillations of the interface lead to the turbulent structure at later times (e.g., $t = 900$s shown in the same figure), typical for the Kelvin-Helmholtz phenomenon. This has been studied intensively by numerical experiments, see, e.g., [18, 34, 37]; note that the exact solution to this problem is unknown.

We compare results obtained by the DEG method with those obtained by the DG method with the Rusanov flux model shown on the right-hand side in Figure 3 for which the isolines exhibit high frequency oscillations, as mentioned above. At later times ($t \geq 700$s) island-like tiny-scale isolated spots (mini-bubbles) develop due to the reconnections between the close located, oscillating isolines. Such structure can be seen for all iso-levels (not only for that one located close to the background level $\theta' = 0$, where the oscillations of the background could yield such artifacts).

A simple increasing of the artificial viscosity in order to damp the oscillations cannot really be considered as a reliable method to eliminate this issue. Although it is known that by increasing the viscosity one can damp the oscillations, we show that the final solution can change on large scales significantly, too. In Figure 4 we show the simulation results using the Rusanov flux model with the constant artificial viscosity $\mu = 0.25\text{m}^2/\text{s}$. This value of the viscosity has been chosen because it is just enough to make the model stable against the short length scale fluctuations and hence to remove the high frequency oscillations. However, this value of μ is already so high that the obtained solution looks very different, if compared to the previously discussed results for $\mu = 0.1\text{m}^2/\text{s}$: i) the Kelvin-Helmholtz instability along the outer interface has been completely damped, and ii) the solution became very dissipative which can be deduced from the shape of the isolines for $\theta' = 0.4$ (the red curve in Figure 4). This isoline shrunk in length for times $t = 600$s and $t = 700$s and it has completely disappeared for later times that means a significantly decreased amplitude of the perturbation, if compared to $\mu = 0.1\text{m}^2/\text{s}$ in Figure 3.

In Figure 5 we show a comparison of the solutions obtained by the DEG model for several values of the viscosity parameter $\mu \in \{0.1, 0.18, 0.25\}\text{m}^2/\text{s}$ and by the Rusanov flux model with the viscosity parameter $\mu = 0.25\text{m}^2/\text{s}$ chosen for stability reasons. The results of both models agree very well in some regions of the computational domain, e.g., along the interfaces located in the interior of the air bubble where the shear forces are negligible. The outer interface calculated by the DEG method shows significant difference in the details of the simulated structure for the low value of the artificial viscosity parameter ($\mu = 0.1\text{m}^2/\text{s}$) and it approaches the solution of the Rusanov model as the viscosity parameter increases. We would like to point out that the DEG solution remains stable for different choices of the constant viscosity parameter, whereas for the Rusanov model either higher constant viscosity or adaptive viscosity (4.3) has to be applied to stabilize the solution.

In what follows we compare numerical solutions obtained by the DEG method and some DG method using a numerical flux based on the approximate Riemann solution. In particular, we have used the so-called HLLC numerical flux. The HLLC is based on the original idea of Harten, Lax and van Leer [19], that gives a close form of the approximate Riemann solver for hyperbolic conservation laws assuming a two-waves configuration. A modification of the HLL flux has been proposed by Toro, Spruce and Speares, cf. [48], which takes also the intermediate contact wave into account and consequently yields a more accurate Riemann solver. For a detailed formulation of the HLLC Riemann solver we refer the reader to a nice overview book by Toro [48]. We would like to point out that it follows from the construction of the DEG method that for one-dimensional problems the evolution operator is comparable with the approximate Riemann solvers (such as the Roe flux or HLLC flux). To make the problem setting more complex we have modified the initial data (4.1) in such a way, that the data are now discontinuous being $\theta'_c =$ constant inside a ball and 0 otherwise. Our extensive numerical experiments indicate some differences between the solution obtained by the DEG method and the DG method with the HLLC flux, that are visible, in particular, for later time instances, $t = 600$ and $t = 900$. On the other hand, we can clearly see that the differences between both solutions are less dominant as it was in the case of the DG method with the Rusanov flux. In Figure 6 we present a representative comparative result. We should also mention that the results were computed without any mesh adaptivity and using no artificial viscosity.

4.2 Test 2: Collision of a large warm bubble with a small cold bubble

Robert [37] proposed the experiment shown in Figure 7. The shape of the rising warm bubble is affected by the small cold bubble, which slides downwards along the right-hand side of the interface, destroying the symmetry of the warm bubble. Two bubbles are initially placed at $(x_c, y_c) = (500\text{m}, 300\text{m})$ and $(x_c, y_c) = (560\text{m}, 640\text{m})$, for the warm and cold bubbles, respectively (cf. Figure 7). The maximal amplitudes of the initial potential temperature perturbation are $\theta'_c = 0.5\,\text{K}$ and $\theta'_c = -0.15\,\text{K}$, respectively. The profiles of the initial perturbation for the excess potential temperature are given by a Gaussian distribution

$$\theta' = \begin{cases} \theta'_c & \text{for} \quad r \leq r_c \\ \theta'_c \exp\left[-\left(r - r_c\right)^2 / 50^2\right] & \text{for} \quad r > r_c \end{cases}$$

with a flat core of radius $r_c = 150\text{m}$ for the warm bubble and $r_c = 0$ for the cold bubble.

In the previous tests we have seen that one can obtain a stable and an oscillating solution on an adaptive mesh, depending on the value of the viscosity. A high viscosity stabilizes the solution, but changes it too much. Here we perform the simulations with both: on a regular static and on an adaptive mesh, with a constant and an adaptive viscosity in order to study the impact of these approaches on the stability of the methods. The integration time step is set to $\Delta t = 0.1\text{s}$.

We start with the static regular grid of the resolution level $n = 8$ and an adaptive artificial

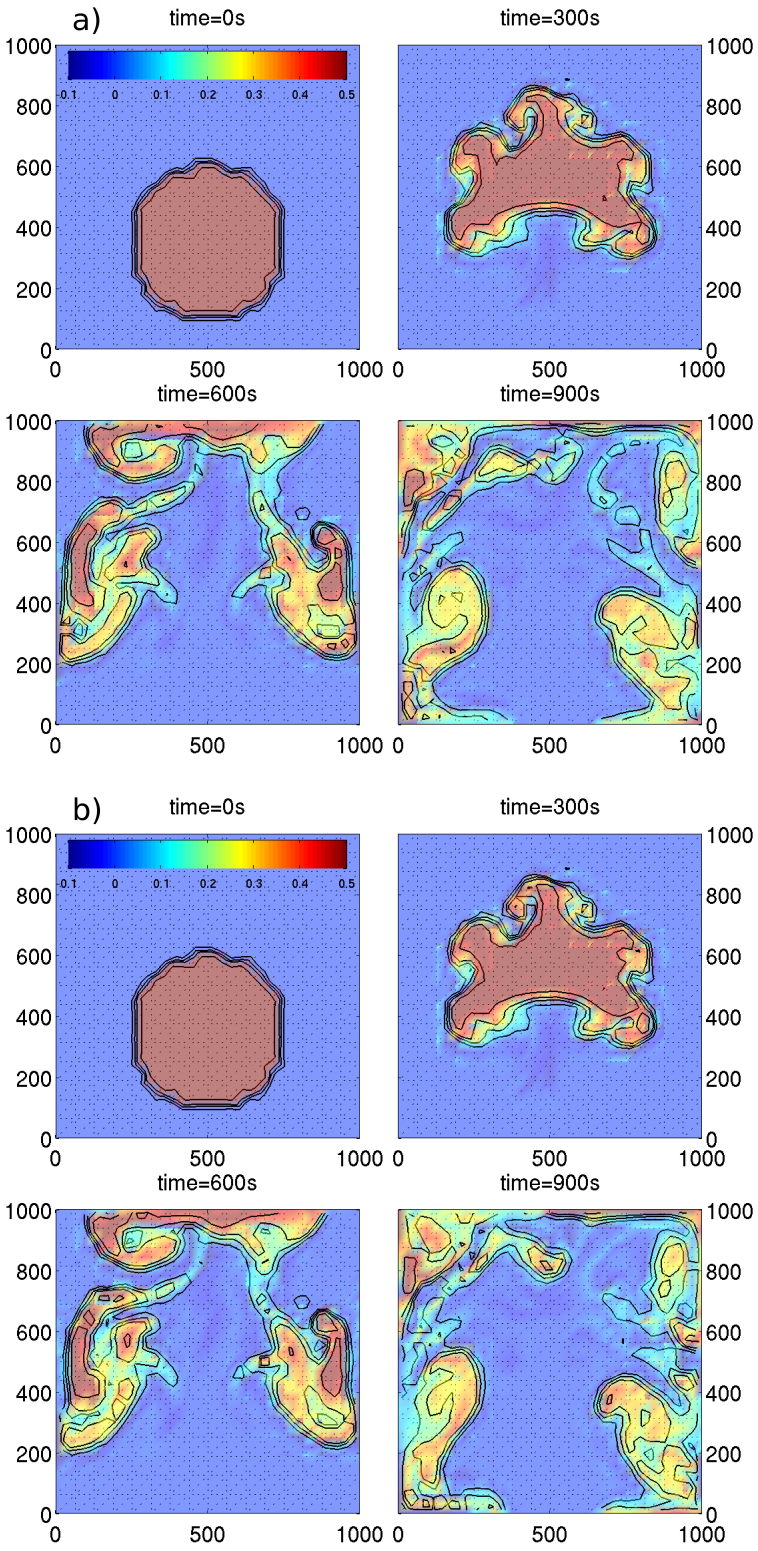

Figure 6: Giraldo-Restelli test with discontinuous initial data. Comparison of the numerical solution obtained by the DG method with the HLLC numerical flux (a) and by DEG method (b). Numerical solution has been computed on mesh with the resolution level $n = 7$ corresponding to the spatial resolution of 31.25m along the shortest element edge (62.5m). $CFL \approx 5.62$, $CFL_{\mathrm{u}} \approx 0.066$ (DEG) and $CFL_{\mathrm{u}} \approx 0.063$ (HLLC).

viscosity in the source term, as mentioned in Section 3. We follow the approach of [34], [46] to stabilize the simulations through the adaptive artificial viscosity calculated from

$$\mu_e = \max\left(\mu_0, \mu_{\text{ref}} \frac{\Delta\theta'_e}{\alpha \Delta\theta'_0} 2^{(12-n)/2}\right), \tag{4.3}$$

where μ_0 is the constant viscosity parameter given by the test case, $\mu_{\text{ref}} = (1/17.7)\text{m}^2/\text{s}$ and $\alpha = 0.4$ are the test independent empirical parameters, $\Delta\theta'_{\text{e}}$ is the maximal deviation of the perturbation θ' on element e, $\Delta\theta'_0$ is $\Delta\theta'_{\text{e}}$ at time $t = 0$. Note: the viscosity μ_e is constant within each element e, but is non-constant for different elements. For more details we refer to [34].

The results for these simulations are shown in Figure 7 a) for the new DEG method and b) for the DG method with the Rusanov flux function. Numerical experiments clearly indicate that the DEG method yields a more stable solution. The solution obtained by the Rusanov flux model exhibits oscillations near the background temperature ($\bar{\theta} = 300\text{K}$), which are not present in the simulations obtained when using the new multidimensional evolution Galerkin operator. To get rid of these oscillations in the Rusanov model we increased the artificial viscosity threshold μ_0 from 0.1 to $0.7\text{m}^2/\text{s}$ (cf. b)-d) in Figure 8). Note that this can influence large scale structures of the solution, too (cf. Test 1).

In the adaptive viscosity approach, the local viscosity on each finite element scales according to (4.3) and for strong temperature gradients it can achieve large values. To understand the behaviour of the solution for a certain given value of the viscosity parameter we compare in Figure 9 the DEG and the Rusanov model for fixed values of μ. In a fully inviscid flow regime (when $\mu = 0$) and up to $\mu = 0.1$ the DEG model yields very reasonable results, stable against the small scale oscillations, although some tiny island-like areas can be observed around the outer interfaces, if compared with the adaptive artificial viscosity results in Figure 8 a). Hence, large scale structures can be studied even for a very low artificial viscosity and we can analyse how small scale oscillations on the outer interface are developing due to the Kelvin-Helmholtz instability. In the same viscosity regime the results obtained by the Rusanov model are strongly oscillating (not shown here) and we had to use a larger adaptive viscosity threshold parameter μ_0 in (4.3) to obtain stable results, see Figures 9 c) and d).

Finally, in Figure 10 we compare the DEG model and the Rusanov model using both the adaptive viscosity and the adaptive mesh. The adaptive viscosity threshold is $\mu_0 = 0.1\text{m}^2/\text{s}$ and the mesh resolution level changes between $n = 1$ and $n = 11$. The DEG method yields a slightly smoother solution. The outer interfaces for the Rusanov model exhibit some oscillations on long-wave lengths, but both solutions are comparable. The tiny island-like areas along the outer interface on the right-hand side is the trace of the cold bubble. This experiment clearly demonstrates that mesh adaptivity is capable of additional adaptive numerical viscosity and stabilization of a numerical solution.

We further analyse the convergence of the numerical scheme toward the exact solution when the grid resolution becomes finer by the experimental order of convergence (EOC)

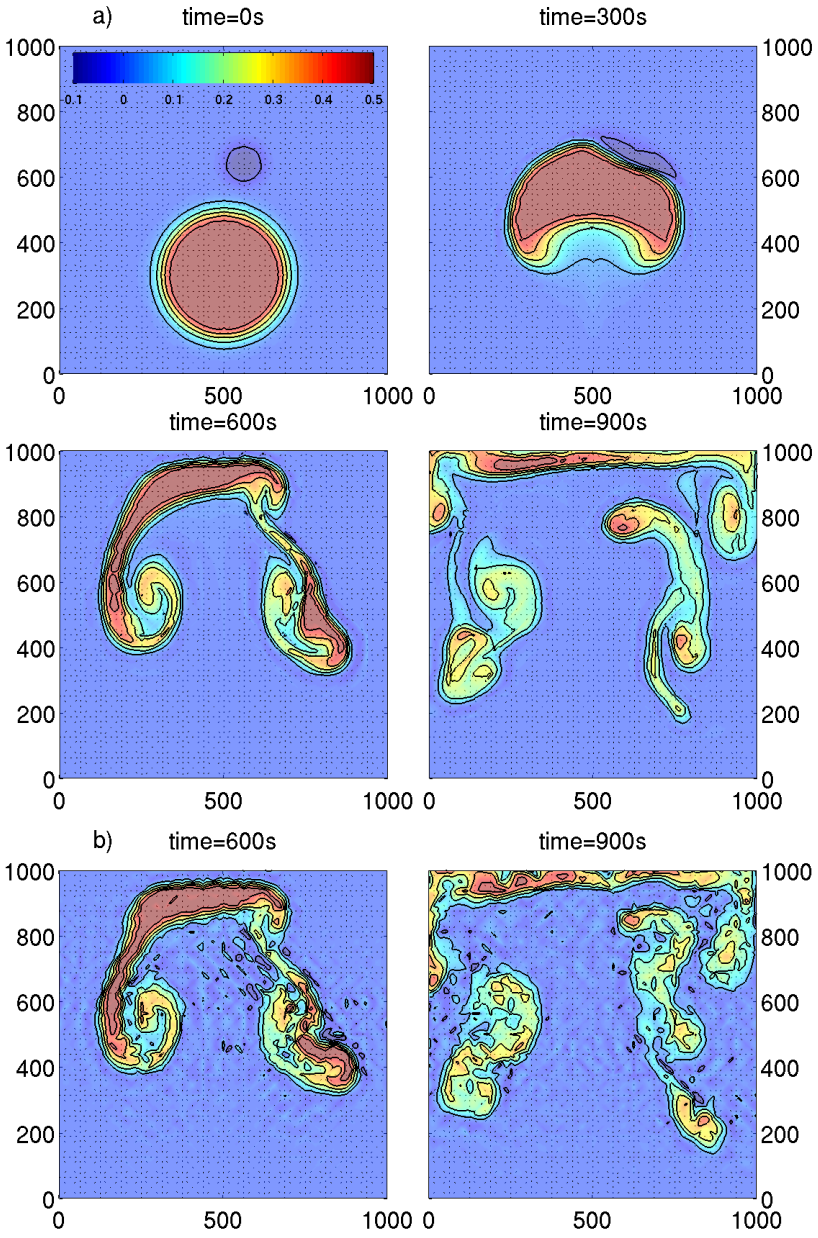

Figure 7: Excess potential temperature θ' for the Robert test [37], obtained by the new semi-implicit method and the adaptive artificial viscosity according to (4.3) with $\mu_0 = 0.1\text{m}^2/\text{s}$: a) multidimensional evolution Galerkin operator for the numerical flux function (DEG); b) DG method with the Rusanov flux function. The real-world domain is 1km×1km, mesh is regular, the resolution level $n = 8$, the spatial resolution ≈ 22.1m, the shortest element edge ≈ 44.2m. The simulation times are as indicated. Note: for the Rusanov flux the snapshots are shown for the last two times only, where the oscillation of the background becomes clearly noticeable. Contour levels correspond to $\theta' \in \{-0.05, 0.05, 0.15, 0.25, 0.35, 0.45\}$K. The integration time step $\Delta t = 0.1$s corresponds to $CFL \approx 1.58$, advective $CFL_{\text{u}} \in \{0, 0.0117, 0.0123, 0.0146\}$ in a) and $CFL_{\text{u}} \in \{0.0123, 0.0173\}$ in b), $CFL_{\text{EG}} \approx 0.0197$.

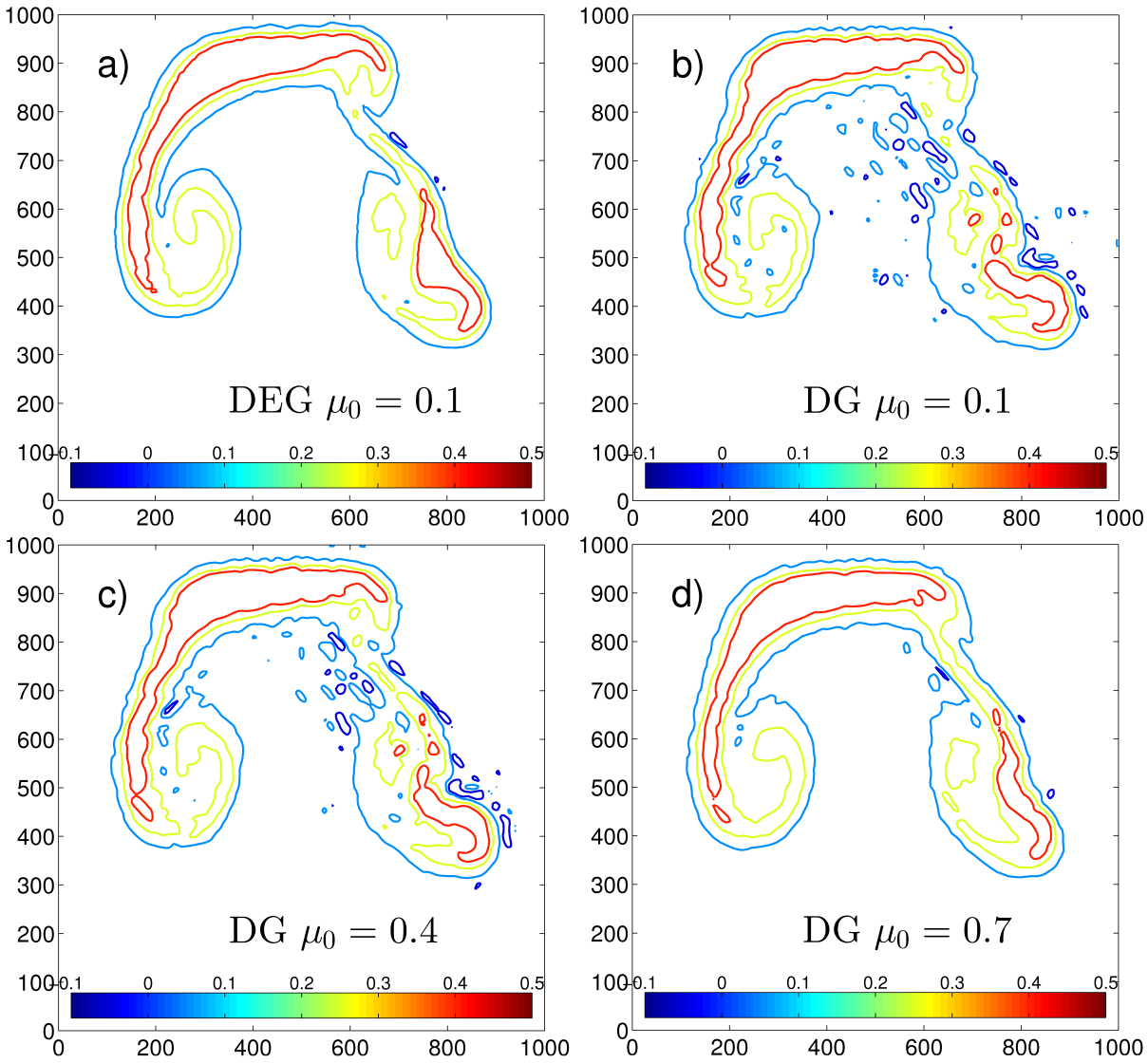

Figure 8: Robert test with the adaptive viscosity according to (4.3): a) DEG method with $\mu_0 = 0.1\text{m}^2/\text{s}$; b)-d) DG method with the Rusanov flux function and $\mu_0 \in \{0.1, 0.4, 0.7\}\text{m}^2/\text{s}$, respectively. Mesh is regular, the resolution level $n = 8$, the spatial resolution ≈ 22.1m, the shortest element edge ≈ 44.2m. Contour curves are for $\theta' \in \{-0.05, 0.05, 0.25, 0.4\}$K, $CFL \approx 1.57$, advective $CFL_{\text{u}} \in \{0.0123, 0.0123, 0.0122, 0.0121\}$ in a)-d). Simulation time $t = 600$s.

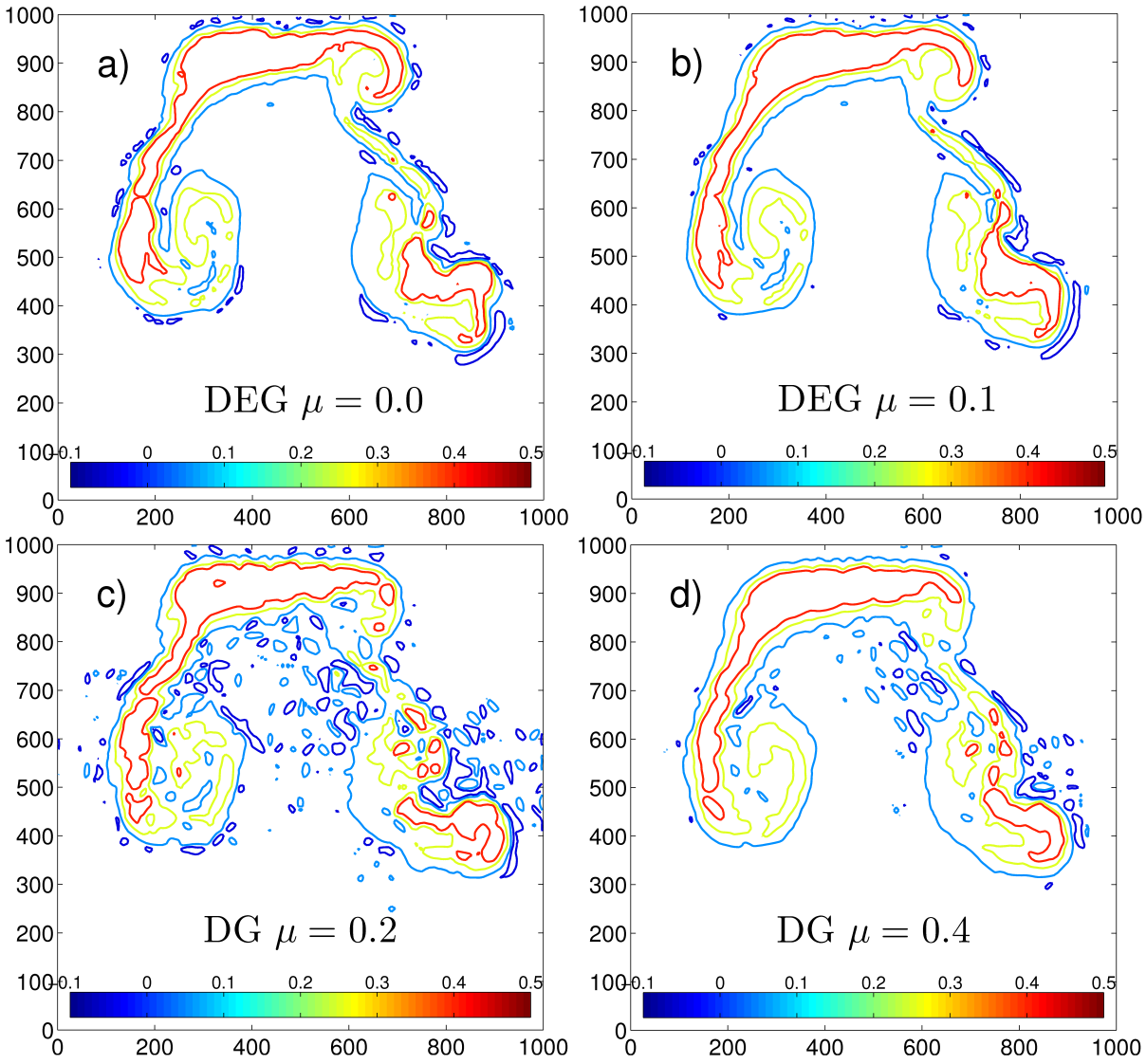

Figure 9: Robert test with constant viscosity: upper row) DEG method with $\mu = 0$ and $0.1\text{m}^2/\text{s}$; bottom row DG method with the Rusanov flux function and $\mu = 0.2$ and $0.4\text{m}^2/\text{s}$. Mesh is regular, the resolution level $n = 8$, the spatial resolution ≈ 22.1m, the shortest element edge ≈ 44.2m. Contour curves are for $\theta' \in \{-0.05, 0.05, 0.25, 0.4\}$K, $CFL \approx 1.57$, advective $CFL_{\text{u}} \in \{0.0132, 0.0127, 0.0127, 0.0122\}$ in a)-d), $CFL_{\text{EG}} \approx 0.0197$. Simulation time $t = 600$s.

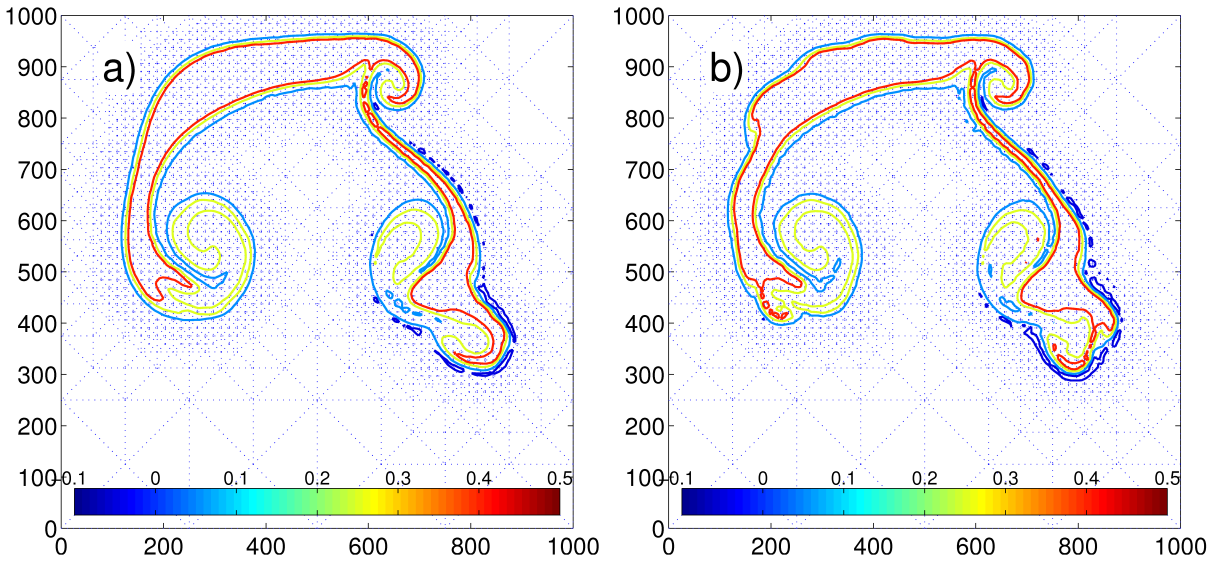

Figure 10: Robert test on the $h-$adaptive mesh with adaptive viscosity according to (4.3) with $\mu_0 = 0.1\text{m}^2/\text{s}$ for the DEG method a) and the DG method with the Rusanov flux b). The coarse/fine mesh resolution is $n = 1 - 11$, corresponding to the spatial resolution of ≈ 7.8m, the shortest element edge ≈ 15.6m. Contour curves are for $\theta' \in \{-0.05, 0.05, 0.25, 0.4\}$K; $CFL \approx 4.45$, advective $CFL_{\text{u}} \approx 0.0378$ (DEG) and $CFL_{\text{u}} \approx 0.0379$ (Rusanov), $CFL_{\text{EG}} \approx 0.055$. Simulation time $t = 600$s.

$$EOC = \log_2 \frac{\|u_n - u\|_{L_1}}{\|u_{n+1} - u\|_{L_1}},$$

where u is an exact solution, u_n and u_{n+1} are the solutions obtained on grids with the resolution levels n and $n+1$, respectively. Since the exact solution is unknown in our experiments we take instead the solutions obtained on the grid with higher resolution and calculate the error between the two solutions and EOC for the DEG and the Rusanov flux methods. They are shown in Table 1. The solution errors for the DEG model are always lower than for the Rusanov model, approximately by a factor of $1.5 - 2$. The EOC of the DEG model is lower than in the Rusanov model on very coarse grids and becomes higher on fine resolution grids.

4.3 Test 3: Density current caused by falling cold air bubble

In the density current experiment proposed by Straka et al. [46], the cold air bubble is placed at $(x_{\text{c}}, y_{\text{c}}) = (0\text{m}, 3000\text{m})$ in the computational domain of 25.6km×6.4km, with the maximal initial temperature perturbation $\Delta T_{\text{c}} = -15\,\text{K}$ and the distribution of the initial temperature perturbation according to

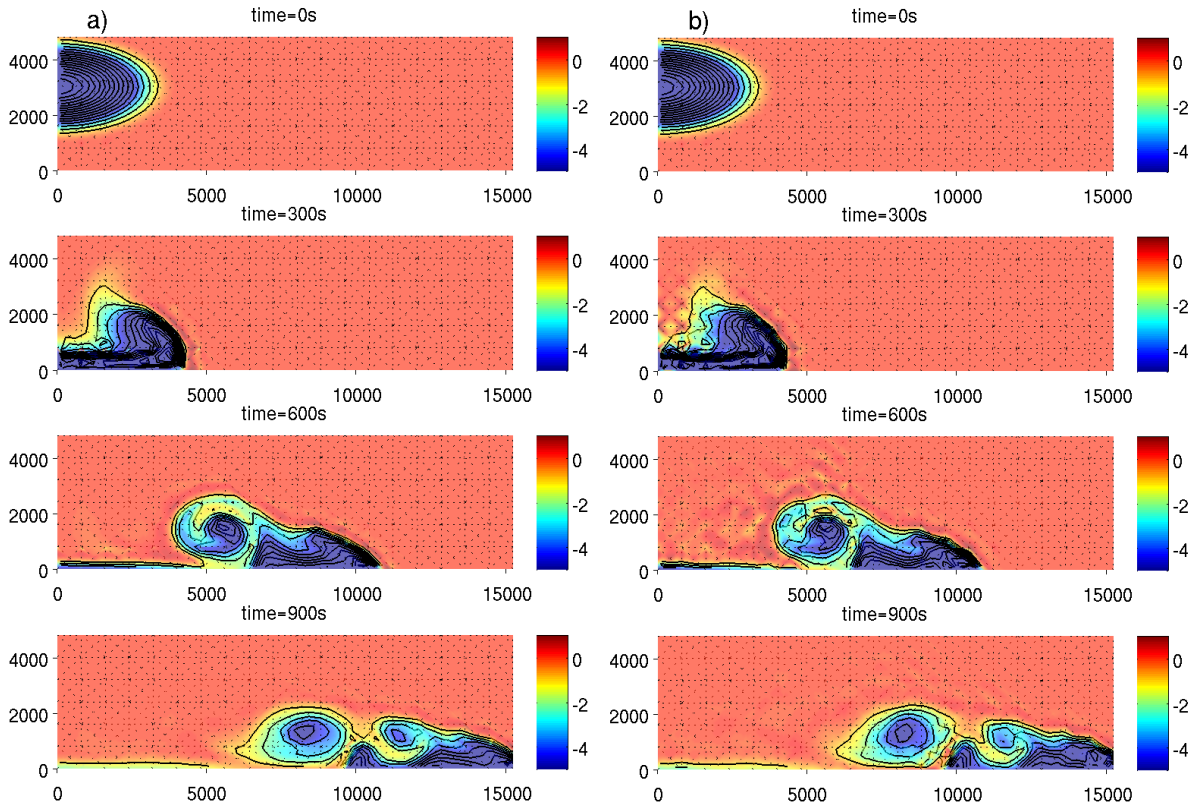

Figure 11: Excess potential temperature θ' for the density current experiment of Straka et al. [46] obtained on a regular mesh: a) multidimensional evolution Galerkin operator for the numerical flux function (DEG); b) DG method with the Rusanov flux function. Artificial viscosity is constant $\mu = 75\mathrm{m}^2/\mathrm{s}$. The real-world domain is 25.6km×6.4km (only part is shown), the mesh resolution level $n = 8$, the spatial resolution ≈ 283m, the finite element shortest edge ≈ 566m. Contour levels correspond to $\theta' = -16$ to -1 by a step of 1 K. Note: the range of the color bar and colors in these snapshots and in Figure 12 has been restricted to -5 K to resolve better the fluctuations near the background $\theta' \approx 0$ K (background potential temperature is $\bar{\theta} = 300$K). The integration time step $\Delta t = 1$s corresponds to $CFL \approx 1.365$, advective $CFL_u \approx 0.141$ in a) and $CFL_u \approx 0.143$ in b), $CFL_{\mathrm{EG}} \approx 1.7 \cdot 10^{-3}$. The simulation times are as indicated.

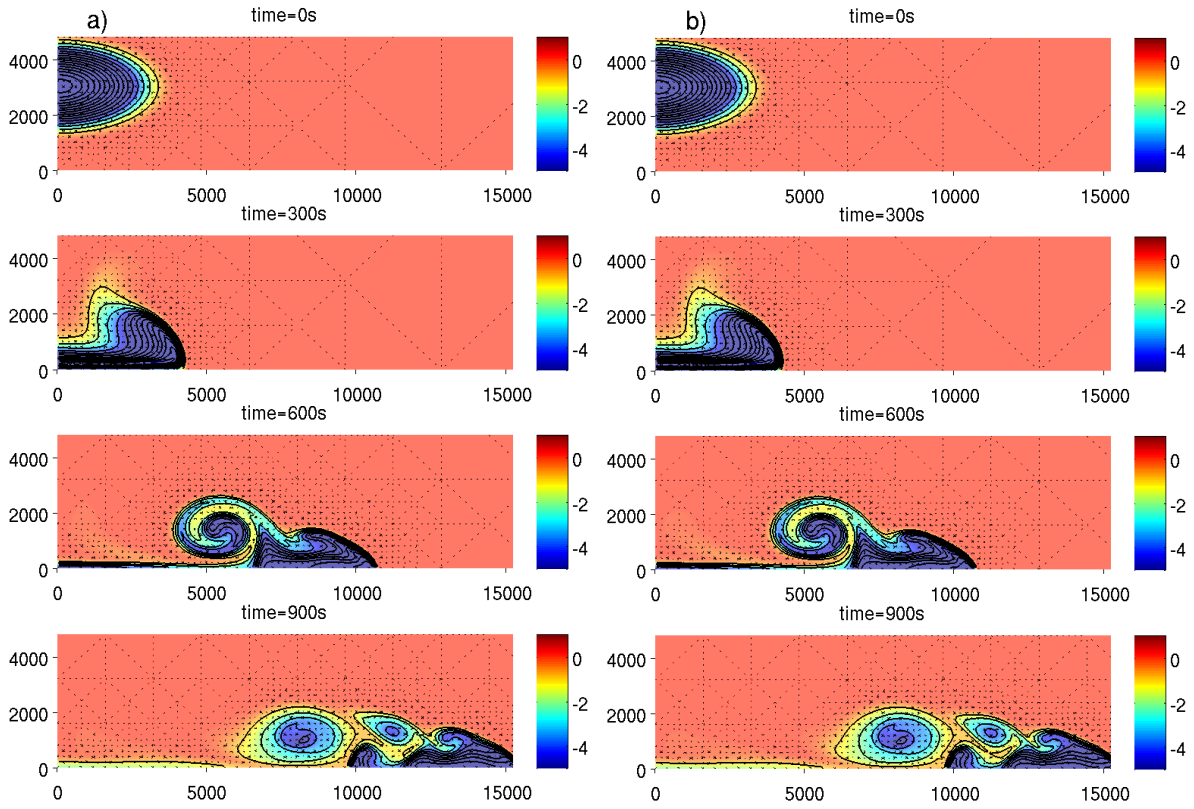

Figure 12: Excess potential temperature θ' for the density current experiment of Straka et al. [46] obtained on the adaptive mesh: a) multidimensional evolution Galerkin operator for the numerical flux function (DEG); b) DG method with the Rusanov flux function. Artificial viscosity is constant $\mu = 75\mathrm{m}^2/\mathrm{s}$. The coarse/fine mesh resolution levels are $n = 1-11$, the finest spatial resolution ≈ 100m, the finite element shortest edge ≈ 200m. Contour levels correspond to $\theta' = -16$ to -1 by a step of 1 K. The integration time step $\Delta t = 0.5$s corresponds to $CFL \approx 1.91$, advective $CFL_u \in \{0,\, 0.194,\, 0.182,\, 0.179\}$ in a) and $CFL_u \in \{0,\, 0.194,\, 0.182,\, 0.178\}$ in b), $CFL_{\mathrm{EG}} \approx 2.4 \cdot 10^{-3}$. The simulation times are as indicated.

a) BDF2+EG:

time = 100s

n = gridlevel	$\|u_n - u_{n+2}\|$/volume	$\|u_{n+2} - u_{n+4}\|$/volume	$EOC(n, n+2, n+4)$
3	0.056854	0.028865	0.9779
4	0.046331	0.019119	1.2769
5	0.028865	0.006807	2.0842
6	0.019119	0.002648	2.8519
7	0.006807	0.000866	2.9740

time = 150s

n = gridlevel	$\|u_n - u_{n+2}\|$/volume	$\|u_{n+2} - u_{n+4}\|$/volume	$EOC(n, n+2, n+4)$
3	0.092453	0.047843	0.9504
4	0.075194	0.030815	1.2870
5	0.047843	0.011797	2.0199
6	0.030815	0.004206	2.8730
7	0.011797	0.001539	2.9383

b) BDF2+Rusanov:

time = 100s

n = gridlevel	$\|u_n - u_{n+2}\|$/volume	$\|u_{n+2} - u_{n+4}\|$/volume	$EOC(n, n+2, n+4)$
3	0.068548	0.033892	1.0162
4	0.059404	0.023547	1.3350
5	0.033892	0.010428	1.7004
6	0.023547	0.004046	2.5522
7	0.010428	0.001697	2.6195

time = 150s

n = gridlevel	$\|u_n - u_{n+2}\|$/volume	$\|u_{n+2} - u_{n+4}\|$/volume	$EOC(n, n+2, n+4)$
3	0.126310	0.065635	0.9445
4	0.107820	0.040578	1.4099
5	0.065635	0.020024	1.7128
6	0.040578	0.007314	2.4719
7	0.020024	0.003247	2.6246

Table 1: The solution error (L_1-norm) and the experimental order of convergence (EOC) in the large time step simulations with a) BDF2+EG and b) BDF2+Rusanov of the Robert test [37] with constant viscosity $\mu = 0.1\mathrm{m}^2/\mathrm{s}$.

$$\Delta T = \begin{cases} 0 & \text{for} \quad r > r_\mathrm{c} \\ (\Delta T_\mathrm{c}/2)\left[1 + \cos\left(\pi r / r_\mathrm{c}\right)\right] & \text{for} \quad r \leq r_\mathrm{c} \end{cases}$$

The initial potential temperature, θ', is calculated from ΔT and the Exner function, $\bar{\pi}$, using the relation $T = \bar{\pi}\theta$.

Since the density of the cold air is higher, the bubble sinks gradually to the bottom of

the simulation domain (negative buoyancy) and continues the viscous motion along the bottom domain boundary. Recall that we have used the no-flux boundary conditions in our simulations. In the test shown in Figure 11 we choose the constant artificial viscosity $\mu = 75\text{m}^2/\text{s}$ from [46] and a regular grid of the resolution level $n = 8$, since we first want to study numerical solutions without any impact of the adaptive mesh on possible instabilities. The time integration was performed with the time step $\Delta t = 1\text{s}$, which corresponds to the following stability condition numbers $CFL \approx 1.365$, $CFL_{\text{u}} \approx 0.141$. For this time step τ has been chosen in such a way that $CFL_{\text{EG}} \approx 1.7 \cdot 10^{-3}$. One can see that the new DEG method reproduces the flow in a slightly more stable way than the one-dimensional Rusanov flux model. At later times, the oscillations near the background temperature are stronger at this (and lower) resolution of the non-adaptive mesh, even though the viscosity is very high in this test. Recall that the background potential temperature has been set to $\bar{\theta} = 300\text{K}$ in our model and the maximal initial potential perturbation is about $\theta'_{\text{c}} = -16.6\,\text{K}$ in this test. When we allow the adaptivity of the mesh and choose a high resolution of the grid, both models yield very similar results, as shown in Figure 12 for the $h-$adaptive grid with coarse/fine resolution levels $n = 1-11$. Here both the DEG and Rusanov flux models perform very well, the differences between the solutions are hardly distinguishable anymore. In this simulation the time step $\Delta t = 0.5\text{s}$ was reduced by a factor of two to adjust it to the finer grid, if compared to the experiment in Figure 11 on the regular mesh $n = 8$.

4.4 Test 4: Inertia-gravity waves

The inertia-gravity waves experiment has been taken from [15]; the initial conditions have been proposed firstly in [43]. The background state is a uniformly stratified atmosphere with a Brunt-Väisälä frequency

$$\mathcal{N} = g\frac{d}{dy}(\ln\bar{\theta})$$

which yields the background potential temperature

$$\bar{\theta} = \theta_0 \exp\left(\frac{\mathcal{N}^2}{g}y\right).$$

The Exner pressure obtained from the hydrostatic balance yields in this case

$$\bar{\pi} = 1 + \frac{g^2}{c_p\theta_0\mathcal{N}^2}\left(\exp\left(-\frac{\mathcal{N}^2}{g}y\right) - 1\right).$$

The perturbation of the potential temperature is given by

$$\theta' = \theta_c \frac{\sin\left(\frac{\pi_c y}{h_c}\right)}{1 + \left(\frac{x-x_c}{a_c}\right)^2},$$

where $\theta_c = 0.01\text{K}$, $\theta_0 = 300\text{K}$, $h_c = 10\text{km}$, $a_c = 5\text{km}$, $x_c = 100\text{km}$, and $\mathcal{N} = 0.01\text{s}^{-1}$. The domain size in (x, y) directions is $300\text{km}\times 10\text{km}$. We use periodic and no-flux boundary

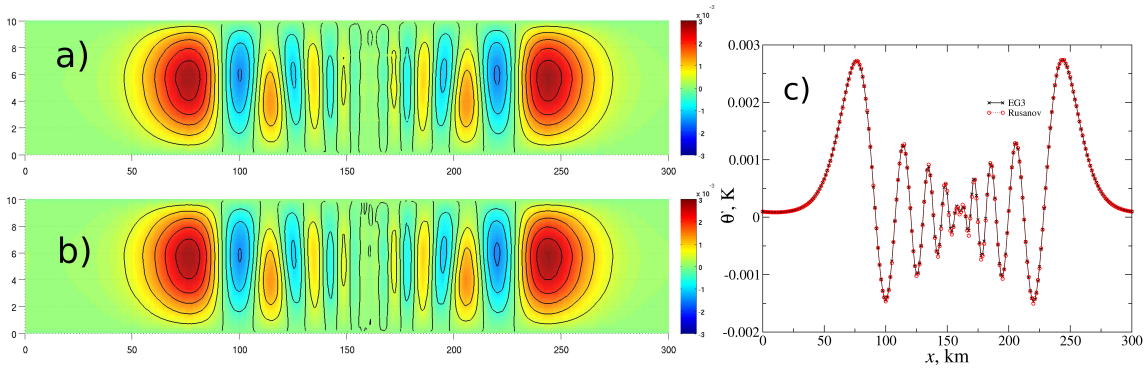

Figure 13: Excess potential temperature for the inertia-gravity waves experiment obtained after time 3000s on a regular mesh with the mesh resolution level $n = 4$: a) multidimensional evolution Galerkin operator for the numerical flux function (DEG); b) DG method with the Rusanov flux function. The real-world domain is 300km×10km, the spatial resolution 1.25km, the finite element shortest edge 2.5km. Contour levels correspond to $\theta' = -0.0015$ to 0.003 by a step of 0.0005 K. The integration time step $\Delta t = 5$s corresponds to $CFL \approx 1.463$ and advective $CFL_u \approx 0.08$ in both tests, $CFL_{\mathrm{EG}} \approx 0.05$. c) Excess potential temperature profiles along the altitude $y = 5$km.

conditions for the horizontal and vertical directions, respectively. This initial state is advected horizontally with a constant velocity $\bar{u} = 20$m/s.

Figure 13 and Figure 14 show the excess potential temperature for the DEG and Rusanov flux models obtained after 3000s on regular grids with the resolution levels $n = 4$ and $n = 2$, respectively. Whilst on the finer mesh the results look very similar, one can see that some differences appear on the coarser grid near the peaks of the potential temperature profiles shown in Figure 14c). The global extreme values of the results on both meshes are similar for both flux models: On the finer mesh ($n = 4$) the results are in the range $\theta' = [-0.001540,\ 0.002774]$ and $v = [-0.002463,\ 0.002665]$ for the DEG, $\theta' = [-0.001568,\ 0.002776]$ and $v = [-0.002468,\ 0.002698]$ for the Rusanov flux. On the coarser mesh ($n = 2$) the results are in the range $\theta' = [-0.001670,\ 0.002787]$ and $v = [-0.002911,\ 0.002723]$ for the DEG flux, $\theta' = [-0.001676,\ 0.002744]$ and $v = [-0.002714,\ 0.002652]$ for the Rusanov flux. We can notice that both the DEG method as well as the DG method with the Rusanov flux give a reasonable approximation of the solution. The results obtained by the DG method with the Rusanov flux however fails to preserve the symmetry of the solution on a coarse grid. Moreover, the results obtained by the DEG method give slightly better approximation of the extrema values, which were reported in [15] for the quadrilateral-based spectral element method using 10th order polynomials.

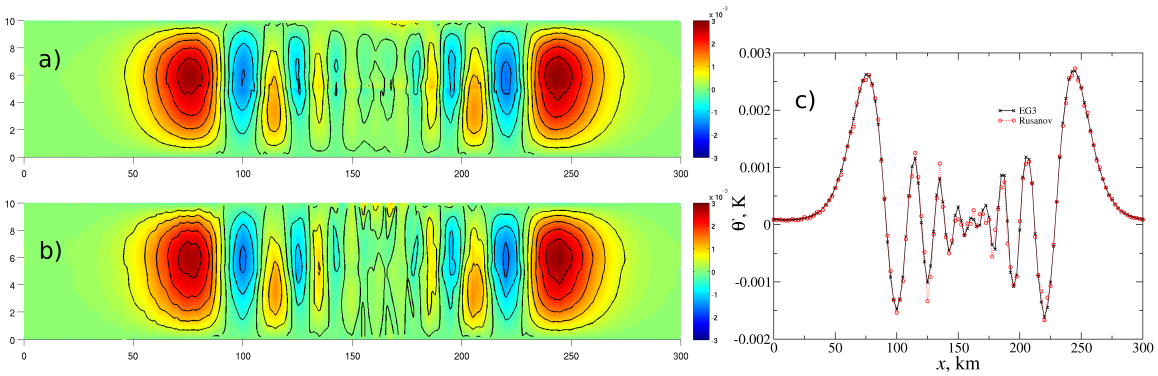

Figure 14: Excess potential temperature for the inertia-gravity waves experiment obtained after time 3000s on a regular mesh with the mesh resolution level $n = 2$: a) multidimensional evolution Galerkin operator for the numerical flux function (DEG); b) DG method with the Rusanov flux function. The real-world domain is 300km×10km, the spatial resolution 2.5km, the finite element shortest edge 5km. Contour levels correspond to $\theta' = -0.0015$ to 0.003 by a step of 0.0005 K. The integration time step $\Delta t = 10$s corresponds to $CFL \approx 1.458$ and advective $CFL_u \approx 0.08$ in both tests, $CFL_{\mathrm{EG}} \approx 0.05$. c) Excess potential temperature profiles along the altitude $y = 5$km.

5 Conclusions

In the present paper we have derived a new adaptive discontinuous evolution Galerkin method. The novelty of our approach relies in the combination of the discontinuous Galerkin method with a genuinely multidimensional evolution operators based on the theory of bicharacteristics for underlying hyperbolic balance laws. In this paper the DEG method is applied for test cases modeling dry atmospheric convection. In order to take into account multiscale phenomena that typically arise in atmospheric flows we split fluxes into a linear part governing acoustic waves and the resulting nonlinear part. The linear operator has to be chosen in such a way that the fastest waves of the system are retained, although in their linearized form. Time integration is realized by the IMEX type approximation using the semi-implicit BDF2 method. In order to efficiently resolve small scale flow structures adaptive mesh refinement is used. This is realized via the AMATOS function library. Numerical experiments presented in Section 4 demonstrate high accuracy, stability and robustness of the new method and illustrate that complex multidimensional flow structures are approximated in a better way than by the discontinuous Galerkin method with the Rusanov flux function. The Rusanov flux model is fast and can yield solutions of similar quality as the DEG model when adaptive mesh refinement and adaptive artificial viscosity are used. On the other hand, the DEG method is more stable due to the truly multidimensional nature of the EG operator. In practice, this implies that less artificial viscosity is required. For low viscosity regimes and for coarse grids, where the Rusanov flux model can be unstable, the DEG model performs better; this is an important finding because atmospheric models should be used with little to no artificial viscosity and they are almost always run completely under-resolved and so methods that

remain stable at these regimes should be preferred. Our first numerical experiments with the DG method using the HLLC numerical flux indicate that the use of one-dimensional approximate Riemann solvers may be another option yielding comparable results. Further increase of the efficiency of the DEG method can be achieved by porting the calculation of the EG operator to graphics processing units (GPU), as has been done recently for the explicit DEG method [7]. In the future it will be interesting to generalize the DEG method for three-dimensional flows and apply it to more complex atmospheric problems, such as simulation of a cloud environment.

Appendix

The aim of this section is to present in more details the derivation of exact integral equations for the nonlinear Euler equations (3.11) presented in Section 3.2. We follow a general strategy as described, e.g., in [29, 30]. As pointed out in Section 3.2 we first linearize (3.8) by freezing the Jacobian matrices at a suitable intermediate states $\tilde{\rho'}, \tilde{u}, \tilde{v}, \tilde{p'}$ and obtain (3.8) with constant Jacobian matrices. Further, we have for the system (3.8) the following eigenvector matrices $\boldsymbol{R}$ and $\boldsymbol{R}^{-1}$

$$\boldsymbol{R}(\mathbf{w}) = \begin{pmatrix} -\frac{\tilde{\rho}}{\tilde{a}} & 1 & 0 & \frac{\tilde{\rho}}{\tilde{a}} \\ n_x & 0 & n_y & n_x \\ n_y & 0 & -n_x & n_y \\ -\tilde{\rho}\,\tilde{a} & 0 & 0 & \tilde{\rho}\,\tilde{a} \end{pmatrix} \quad \boldsymbol{R}^{-1}(\mathbf{w}) = \frac{1}{2} \begin{pmatrix} 0 & n_x & n_y & -\frac{1}{\tilde{\rho}\,\tilde{a}} \\ 2 & 0 & 0 & -\frac{2}{\tilde{a}^2} \\ 0 & 2\,n_y & -2\,n_x & 0 \\ 0 & n_x & n_y & \frac{1}{\tilde{\rho}\,\tilde{a}} \end{pmatrix}.$$

Expressing an arbitrary unit vector $\mathbf{n} = (n_x, n_y)$ as $\mathbf{n} = (\cos(\omega), \sin(\omega))$, $\omega \in (0, 2\pi]$ we can calculate the characteristic vector $\mathbf{v} = \boldsymbol{R}^{-1}\mathbf{w}$

$$\mathbf{v} := \begin{pmatrix} \frac{1}{2}(u\cos(\omega) + v\sin(\omega) - \frac{p'}{\tilde{\rho}\tilde{a}}) \\ \rho' - \frac{p'}{\tilde{a}^2} \\ u\sin(\omega) - v\cos(\omega) \\ \frac{1}{2}(u\cos(\omega) + v\sin(\omega) + \frac{p'}{\tilde{\rho}\tilde{a}}) \end{pmatrix} \tag{5.1}$$

and the right hand side source term vector $\mathbf{r} = \boldsymbol{R}^{-1}\mathbf{s}(\mathbf{w})$

$$\mathbf{r} := \begin{pmatrix} -\frac{1}{2}\sin(\omega) g \frac{\rho'}{\rho} + \frac{1}{2}\frac{v}{\tilde{\rho}\,\tilde{a}}\partial_y \bar{p} \\ -\partial_y \bar{\rho}\, v + \frac{\partial_y \bar{p}}{\tilde{a}^2}\, v \\ \cos(\omega)\, g\, \frac{\rho'}{\rho} \\ -\frac{1}{2}\sin(\omega) g \frac{\rho'}{\rho} - \frac{1}{2}\frac{v}{\tilde{\rho}\,\tilde{a}}\partial_y \bar{p} \end{pmatrix}.$$

The diagonal parts of the transformed Jacobians $\boldsymbol{B}_i$, $i = 1, 2$, read

$$\operatorname{diag}(\boldsymbol{B}_1) = \operatorname{diag}(\boldsymbol{R}^{-1}\boldsymbol{A}_1\boldsymbol{R}) = \begin{pmatrix} \tilde{u} - \cos(\omega)\,\tilde{a} & 0 & 0 & 0 \\ 0 & \tilde{u} & 0 & 0 \\ 0 & 0 & \tilde{u} & 0 \\ 0 & 0 & 0 & \tilde{u} + \cos(\omega)\,\tilde{a} \end{pmatrix}$$

$$\operatorname{diag}(\boldsymbol{B}_2) = \operatorname{diag}(\boldsymbol{R}^{-1}\boldsymbol{A}_2\boldsymbol{R}) = \begin{pmatrix} \tilde{v} - \sin(\omega)\,\tilde{a} & 0 & 0 & 0 \\ 0 & \tilde{v} & 0 & 0 \\ 0 & 0 & \tilde{v} & 0 \\ 0 & 0 & 0 & \tilde{v} + \sin(\omega)\,\tilde{a} \end{pmatrix}$$

and the multidimensional source term $\mathbf{S}$ in (3.10) has the following form

$$\begin{aligned} \mathbf{S} &:= -\left[(\boldsymbol{B}_1 - \operatorname{diag}(\boldsymbol{B}_1))\partial_x \mathbf{v} + (\boldsymbol{B}_2 - \operatorname{diag}(\boldsymbol{B}_2))\partial_y \mathbf{v}\right] \\ &= \begin{bmatrix} \frac{1}{2}\tilde{a}\,(\sin(\omega)\partial_x v_3 - \cos(\omega)\partial_y v_3) \\ 0 \\ \tilde{a}\sin(\omega)(\partial_x v_1 - \partial_x v_4) - \tilde{a}\cos(\omega)(\partial_y v_1 - \partial_y v_4) \\ -\frac{1}{2}\tilde{a}\,(\sin(\omega)\partial_x v_3 - \cos(\omega)\partial_y v_3) \end{bmatrix}. \end{aligned} \tag{5.2}$$

Due to linearization of the system (3.8) the bicharacteristics $\mathbf{x}_j$, $j = 1, \ldots, 4$ are straight lines that create a mantle of the so-called bicharacteristic cone, cf. Figure 1. They connect a peak $\mathbf{P} = (x, y, t_n + \tau)$ of the cone with the corresponding foot points $\mathbf{Q}_j(\omega)$ at time t_n

$$\begin{aligned} \mathbf{Q}_1(\omega) &= [x - (\tilde{u} - \tilde{a}\cos(\omega))\tau,\ y - (\tilde{v} - \tilde{a}\sin(\omega))\tau,\ t_n]^{\mathsf{T}} \\ \mathbf{Q}_2 = \mathbf{Q}_3 &= [x - \tilde{u}\tau,\ y - \tilde{v}\tau,\ t_n]^{\mathsf{T}} \\ \mathbf{Q}_4(\omega) &= [x - (\tilde{u} + \tilde{a}\cos(\omega))\tau,\ y - (\tilde{v} + \tilde{a}\sin(\omega))\tau,\ t_n]^{\mathsf{T}},\ \omega \in (0, 2\pi]. \end{aligned}$$

Realizing that $\mathbf{Q}_1(\omega) = \mathbf{Q}_4(\omega + \pi)$ we get the following symmetry properties

$$\begin{aligned} v_1(\mathbf{Q}_1(\omega + \pi), \omega + \pi) &= -v_4(\mathbf{Q}_4(\omega), \omega), \\ v_1(\mathbf{Q}_1(\omega + \pi), \omega + \pi)\cos(\omega + \pi) &= v_4(\mathbf{Q}_4(\omega), \omega)\cos(\omega), \\ v_1(\mathbf{Q}_1(\omega + \pi), \omega + \pi)\sin(\omega + \pi) &= v_4(\mathbf{Q}_4(\omega), \omega)\sin(\omega). \end{aligned}$$

Since all v_j are 2π periodic we have for the integrals over the perimiter of the cone basis

$$\begin{aligned} \int_0^{2\pi} v_1(\mathbf{Q}_1(\omega), \omega)d\omega &= -\int_0^{2\pi} v_4(\mathbf{Q}_4(\omega), \omega)d\omega, \\ \int_0^{2\pi} v_1(\mathbf{Q}_1(\omega), \omega)\cos(\omega)d\omega &= \int_0^{2\pi} v_4(\mathbf{Q}_4(\omega), \omega)\cos(\omega)d\omega, \\ \int_0^{2\pi} v_1(\mathbf{Q}_1(\omega), \omega)\sin(\omega)d\omega &= \int_0^{2\pi} v_4(\mathbf{Q}_4(\omega), \omega)\sin(\omega)d\omega. \end{aligned} \tag{5.3}$$

Analogously, it holds for the source term

$$\begin{aligned} r_1(\mathbf{x}_1(t,\omega),t,\omega) &= -\frac{1}{2}\sin(\omega)g\frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega)) + \frac{1}{2}\frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}}\partial_y\bar{p} \\ &= \frac{1}{2}\sin(\omega+\pi)g\frac{\rho'}{\rho}(\mathbf{x}_4(t,\omega+\pi)) + \frac{1}{2}\frac{v(\mathbf{x}_4(t,\omega+\pi))}{\tilde{\rho}\,\tilde{a}}\partial_y\bar{p} \\ &= -r_4(\mathbf{x}_4(t,\omega+\pi),t,\omega+\pi) \end{aligned}$$

and consequently

$$\begin{aligned} \int_0^{2\pi}\int_{t_n}^{t_n+\tau} r_1(\mathbf{x}_1(t,\omega),t,\omega)\,dt\,d\omega &= -\int_0^{2\pi}\int_{t_n}^{t_n+\tau} r_4(\mathbf{x}_4(t,\omega),t,\omega)\,dt\,d\omega \\ \int_0^{2\pi}\int_{t_n}^{t_n+\tau} r_1(\mathbf{x}_1(t,\omega),t,\omega)\cos(\omega)\,dt\,d\omega &= \int_0^{2\pi}\int_{t_n}^{t_n+\tau} r_4(\mathbf{x}_4(t,\omega),t,\omega)\cos(\omega)\,dt\,d\omega \\ \int_0^{2\pi}\int_{t_n}^{t_n+\tau} r_1(\mathbf{x}_1(t,\omega),t,\omega)\sin(\omega)\,dt\,d\omega &= \int_0^{2\pi}\int_{t_n}^{t_n+\tau} r_4(\mathbf{x}_4(t,\omega),t,\omega)\sin(\omega)\,dt\,d\omega\,. \end{aligned} \tag{5.4}$$

Denoting $c_j := \int_{t_n}^{t_n+\tau} S_j(\mathbf{x}_j(t,\omega),t,\omega) + r_j(\mathbf{x}_j(t,\omega),t,\omega)\,dt$, $j = 1,\dots,4$, we also have, cf. [30],

$$\begin{aligned} \int_0^{2\pi} c_1(\mathbf{x}_1(\omega))\,d\omega &= -\int_0^{2\pi} c_4(\mathbf{x}_4(\omega))\,d\omega \\ \int_0^{2\pi} c_1(\mathbf{x}_1(\omega))\cos(\omega)\,d\omega &= \int_0^{2\pi} c_4(\mathbf{x}_4(\omega))\cos(\omega)\,d\omega \\ \int_0^{2\pi} c_1(\mathbf{x}_1(\omega))\sin(\omega)\,d\omega &= \int_0^{2\pi} c_4(\mathbf{x}_4(\omega))\sin(\omega)\,d\omega\,. \end{aligned} \tag{5.5}$$

As described in Section 3.2 by integrating (3.10) along the corresponding bicharacteristics $\mathbf{x}_j$ from t_n to $t_n+\tau$ we already obtain an exact integral representation of the solution expressed in the characteristic variables $\mathbf{v}$

$$v_j(\mathbf{P}) = v_j(\mathbf{Q}_j(\omega)) + \int_{t_n}^{t_n+\tau} (S_j + r_j)\,(\mathbf{x}_j(t,\omega),t,\omega)\,dt, \quad \omega\in(0,2\pi], \quad j=1,\dots,4\,. \tag{5.6}$$

Multiplying (5.6) by $\boldsymbol{R}$ from left and integrating over a parameter ω from $(0,2\pi]$ we perform the backward transformation and obtain the exact representation in the original

primitive variables

$$
\begin{aligned}
\mathbf{w}(\mathbf{P}) =& \boldsymbol{R}\left[\frac{1}{2\pi}\int_0^{2\pi} v_j(\mathbf{Q}_j(\omega))\,d\omega \right.\\
&\left. +\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau} S_j(\mathbf{x}_j(t,\omega),t,\omega)+r_j(\mathbf{x}_j(t,\omega),t,\omega)dt\;d\omega\right]_{j=1,\dots,4}\\
=&\frac{1}{2\pi}\int_0^{2\pi}\begin{bmatrix} -\frac{\tilde\rho}{\tilde a}v_1+v_2+\frac{\tilde\rho}{\tilde a}v_4\\ \cos(\omega)\,v_1+\sin(\omega)\,v_3+\cos(\omega)\,v_4\\ \sin(\omega)\,v_1-\cos(\omega)\,v_3+\sin(\omega)\,v_4\\ -\tilde\rho\tilde a\,v_1+\tilde\rho\tilde a\,v_4\end{bmatrix}d\omega\\
&+\frac{1}{2\pi}\int_0^{2\pi}\begin{bmatrix} -\frac{\tilde\rho}{\tilde a}c_1+c_2+\frac{\tilde\rho}{\tilde a}c_4\\ \cos(\omega)\,c_1+\sin(\omega)\,c_3+\cos(\omega)\,c_4\\ \sin(\omega)\,c_1-\cos(\omega)\,c_3+\sin(\omega)\,c_4\\ -\tilde\rho\tilde a\,c_1+\tilde\rho\tilde a\,c_4\end{bmatrix}d\omega\,. \qquad (5.7)
\end{aligned}
$$

Now, taking into account the above symmetry properties (5.3), (5.4) and (5.5) we can rewrite (5.7) in the following way

$$
\begin{aligned}
\mathbf{w}(\mathbf{P}) =&\frac{1}{2\pi}\int_0^{2\pi}\begin{bmatrix} -2\frac{\tilde\rho}{\tilde a}v_1(\mathbf{Q}_1(\omega))+v_2(\mathbf{Q}_2(\omega))\\ 2\cos(\omega)\,v_1(\mathbf{Q}_1(\omega))+\sin(\omega)\,v_3(\mathbf{Q}_3(\omega))\\ 2\sin(\omega)\,v_1(\mathbf{Q}_1(\omega))-\cos(\omega)\,v_3(\mathbf{Q}_3(\omega))\\ -2\tilde\rho\tilde a\,v_1(\mathbf{Q}_1(\omega))\end{bmatrix}d\omega\\
&+\frac{1}{2\pi}\int_0^{2\pi}\begin{bmatrix} -\frac{2\tilde\rho}{\tilde a}\,c_1(\mathbf{x}_1(\omega))+c_2(\mathbf{x}_2(\omega))\\ 2\cos(\omega)\,c_1(\mathbf{x}_1(\omega))+\sin(\omega)\,c_3(\mathbf{x}_3(\omega))\\ 2\sin(\omega)\,c_1(\mathbf{x}_1(\omega))-\cos(\omega)\,c_3(\mathbf{x}_3(\omega))\\ -2\tilde\rho a\,c_1(\mathbf{x}_1(\omega))\end{bmatrix}d\omega\,. \qquad (5.8)
\end{aligned}
$$

Plugging (5.1) into (5.8) and realizing that the second and third characteristics $\mathbf{x}_2=\mathbf{x}_3$ are independent on the angle ω, $\mathbf{Q}_2=[x-\tilde u\tau, y-\tilde v\tau, t_n]^{\mathsf T}=\mathbf{Q}_3$, we have, for example, for the first component $w_1(\mathbf{P})=\rho'(\mathbf{P})$ the following exact integral representation

$$
\begin{aligned}
\rho'(\mathbf{P}) =& \frac{\tilde\rho}{2\pi\tilde a}\int_0^{2\pi}\left[-\cos(\omega)\,u(\mathbf{Q}_1(\omega))-\sin(\omega)\,v(\mathbf{Q}_1(\omega))+\frac{1}{\tilde\rho\,\tilde a}p'(\mathbf{Q}_1(\omega))\right]d\omega\\
&+\rho'(\mathbf{Q}_2)-\frac{p'(\mathbf{Q}_2)}{\tilde a^2}+\frac{1}{2\pi}\int_0^{2\pi}\left[-\frac{2\,\tilde\rho}{\tilde a}c_1(\mathbf{x}_1(\omega))+c_2(\mathbf{x}_2)\right]d\omega. \qquad (5.9)
\end{aligned}
$$

Finally, we need to express the mantle integrals of the source terms $c_j=\int_{t_n}^{t_n+\tau} S_j(\mathbf{x}_j(t,\omega),t,\omega)+$

$r_j(\mathbf{x}_j(t,\omega),t,\omega)\,dt$, $j=1,\dots,4$, i.e.

$$\int_0^{2\pi}\int_{t_n}^{t_n+\tau} c_1\,dt\,d\omega\ ,\quad \int_0^{2\pi}\int_{t_n}^{t_n+\tau} \cos(\omega)c_1\,dt\,d\omega\ ,\quad \int_0^{2\pi}\int_{t_n}^{t_n+\tau} \sin(\omega)c_1\,dt\,d\omega\ ,$$
$$\int_0^{2\pi}\int_{t_n}^{t_n+\tau} c_2\,dt\,d\omega\ ,\quad \int_0^{2\pi}\int_{t_n}^{t_n+\tau} \cos(\omega)\,c_3\,dt\,d\omega\ ,\quad \int_0^{2\pi}\int_{t_n}^{t_n+\tau} \sin(\omega)\,c_3\,dt\,d\omega.$$

First, plugging (5.1) in (5.2) the source term $\mathbf{S}$ can be written as

$$\mathbf{S}(\mathbf{w}) = \begin{bmatrix} \frac{1}{2}\tilde{a}[\partial_x u\sin^2(\omega)+\partial_y v\cos^2(\omega)-(\partial_y u+\partial_x v)\sin(\omega)\cos(\omega)] \\ 0 \\ \frac{1}{\tilde{\rho}}(\cos(\omega)\,\partial_y p'-\sin(\omega)\,\partial_x p') \\ -\frac{1}{2}\tilde{a}[\partial_x u\sin^2(\omega)+\partial_y v\cos^2(\omega)-(\partial_y u+\partial_x v)\sin(\omega)\cos(\omega)] \end{bmatrix}. \tag{5.10}$$

The second and third bicharacteristics $\mathbf{x}_2(t)$ and $\mathbf{x}_3(t)$ are independent on ω and we can express directly the integrals of source terms c_2 and c_3

$$\begin{aligned}
\int_0^{2\pi}\int_{t_n}^{t_n+\tau} c_3\sin(\omega)\,dt\,d\omega &= \int_0^{2\pi}\int_{t_n}^{t_n+\tau}\frac{1}{\tilde{\rho}}(\cos(\omega)\,\partial_y p'-\sin(\omega)\,\partial_x p')\sin(\omega)\\
+\cos(\omega)\sin(\omega)g\frac{\rho'}{\rho}\,dt\,d\omega &= -\int_{t_n}^{t_n+\tau}\frac{1}{\tilde{\rho}}\partial_x p'\int_0^{2\pi}\sin^2(\omega)\,dt\,d\omega = -\frac{\pi}{\tilde{\rho}}\int_{t_n}^{t_n+\Delta t}\partial_x p'\,dt\ ,\\
\int_0^{2\pi}\int_{t_n}^{t_n+\tau} c_3\cos(\omega)\,dt\,d\omega &= \frac{\pi}{\tilde{\rho}}\int_{t_n}^{t_n+\tau}\partial_y p'\,dt+\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\cos^2(\omega)g\frac{\rho'}{\rho}\,dt\,d\omega\\
&= \frac{\pi}{\tilde{\rho}}\int_{t_n}^{t_n+\tau}\partial_y p'\,dt+\pi g\int_{t_n}^{t_n+\tau}\frac{\rho'}{\rho}\,dt.
\end{aligned} \tag{5.11}$$

$$\int_0^{2\pi}\int_{t_n}^{t_n+\tau} c_2\,dt\,d\omega = 2\pi\int_{t_n}^{t_n+\tau} v(\mathbf{x}_2(t))\left(-\partial_y\bar{\rho}+\frac{\partial_y\bar{p}}{\tilde{a}^2}\right)dt\,. \tag{5.12}$$

Substituting (5.10), (5.11), (5.12) in (5.8) and denoting

$$\begin{aligned}
\beta(t,\omega) &:= S_1(\mathbf{x}_1(t,\omega),t,\omega)-S_4(\mathbf{x}_4(t,\omega),t,\omega)\\
&= \tilde{a}\,[\partial_x u\,\sin^2(\omega)-(\partial_y u+\partial_x v)\sin(\omega)\cos(\omega)+\partial_y v\,\cos^2(\omega)]\,.
\end{aligned}$$

we finally obtain the exact integral representation in primitive variables (ρ',u,v,p')

$$\begin{aligned}
\rho'(\mathbf{P}) = \frac{\tilde{\rho}}{2\pi\tilde{a}}\int_0^{2\pi}&\left[-\cos(\omega)\,u(\mathbf{Q}_1(\omega))-\sin(\omega)\,v(\mathbf{Q}_1(\omega))+\frac{1}{\tilde{\rho}\,\tilde{a}}p'(\mathbf{Q}_1(\omega))\right]d\omega \\
&+\rho'(\mathbf{Q}_2)-\frac{p'(\mathbf{Q}_2)}{\tilde{a}^2}\\
&-\frac{\tilde{\rho}}{2\pi\tilde{a}}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\beta(t,\omega)\,dt\,d\omega\\
&-\frac{\tilde{\rho}}{2\pi\tilde{a}}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}-\sin(\omega)g\frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega))+\frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}}\partial_y\bar{p}\,dt\,d\omega
\end{aligned} \tag{5.13}$$

$$+\int_{t_n}^{t_n+\tau} v(\mathbf{x}_2(t))\left(-\partial_y\bar{\rho}+\frac{\partial_y\bar{p}}{\tilde{a}^2}\right)\,dt\,,$$

$$\begin{aligned}
u(\mathbf{P}) = \frac{1}{2\pi}\int_0^{2\pi}&\left[-\frac{p'(\mathbf{Q}_1(\omega))}{\tilde{\rho}\tilde{a}}\cos(\omega)+u(\mathbf{Q}_1(\omega))\cos^2(\omega)+v(\mathbf{Q}_1(\omega))\sin(\omega)\cos(\omega)\right]\,d\omega\\
&+\frac{1}{2}u(\mathbf{Q}_2)+\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\cos(\omega)\,\beta(t,\omega)\,dt\,d\omega\\
&+\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\cos(\omega)\frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}}\partial_y\bar{p}-\sin(\omega)\cos(\omega)g\frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega))\,dt\,d\omega\\
&-\frac{1}{2\tilde{\rho}}\int_{t_n}^{t_n+\tau}\partial_x p'(\mathbf{x}_2(t))\,dt\,,\\
v(\mathbf{P}) = \frac{1}{2\pi}\int_0^{2\pi}&\left[-\frac{p'(\mathbf{Q}_1(\omega))}{\tilde{\rho}\tilde{a}}\sin(\omega)+u(\mathbf{Q}_1(\omega))\cos(\omega)\sin(\omega)+v(\mathbf{Q}_1(\omega))\sin^2(\omega)\right]\,d\omega\\
&+\frac{1}{2}v(\mathbf{Q}_2)+\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\sin(\omega)\,\beta(t,\omega)\,dt\,d\omega\\
&+\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\sin(\omega)\frac{v(\mathbf{x}_1(t,\omega))}{\tilde{\rho}\,\tilde{a}}\partial_y\bar{p}-\sin^2(\omega)g\frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega))\,dt\,d\omega\\
&-\frac{1}{2\tilde{\rho}}\int_{t_n}^{t_n+\tau}\partial_y p'(\mathbf{x}_2(t))\,dt-\frac{1}{2}g\int_{t_n}^{t_n+\tau}\frac{\rho'}{\rho}(\mathbf{x}_2(t))\,dt\,,\\
p'(\mathbf{P}) = \frac{1}{2\pi}\int_0^{2\pi}&\left[p'(\mathbf{Q}_1(\omega))-\tilde{\rho}\tilde{a}u(\mathbf{Q}_1(\omega))\cos(\omega)-\tilde{\rho}\tilde{a}v(\mathbf{Q}_1(\omega))\sin(\omega)\right]\,d\omega\\
&-\tilde{\rho}\tilde{a}\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}\beta(t,\omega)\,dt\,d\omega\\
&-\frac{1}{2\pi}\int_0^{2\pi}\int_{t_n}^{t_n+\tau}-\sin(\omega)\tilde{\rho}\tilde{a}\,g\frac{\rho'}{\rho}(\mathbf{x}_1(t,\omega))+v(\mathbf{x}_1(t,\omega))\partial_y\bar{p}\,dt\,d\omega.
\end{aligned}$$

Acknowledgments

This research has been supported by the German Research Foundation DFG under the grants LU 1470/2-2, LU 1470/2-3 and the priority program MetStröm (SPP 1276). ML, VW and LY further acknowledge the support of the Center of Computational Sciences in Mainz. AM and FXG gratefully acknowledge the support of the National Science Foundation through grant number 1216700 and ONR grant PE-0602435N. ML and LY would like to thank G. Bispen (Mainz) for fruitful discussions on the topic.

References

[1] Arun, K.R., Kraft, M., Lukáčová-Medviďová, M., Prasad, Phoolan Finite volume evolution Galerkin method for hyperbolic conservation laws with spatially varying flux functions, *J. Comp. Phys.* **228**(2) (2009), 565-590.

[2] Noelle, S., Bispen, G., Arun, K.R., Lukáčová-Medviďová, M., Munz, C.-D. Asymptotic preserving all Mach number scheme for the Euler equations of gas dynamics, submitted 2013.

[3] Bacon, D.P., Ahmad, N.N., Boybeyi, Z., Dunn, T.J., Hall, M.S., Lee, P.C.S., Sarma, R.A., Turner, M.D., Waight, K.T., Young, S.H., Zack, J.W. A dynamically adapting weather and dispersion model: The operational multiscale environment model with grid adaptivity (omega), *Mon. Weather Rev.* **128** (2000), 2044-2076.

[4] Bassi, F., Rebay, S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations, *J. Comput. Phys.* **131** (2) (1997), 267-279.

[5] Behrens, J., Rakowsky, N., Hiller, W., Handorf, D., Lauter, M., Papke, J., Dethloff, K. Parallel adaptive mesh generator for atmospheric and oceanic simulation, *Ocean Model* **10** (2005), 171-183.

[6] Behrens, J. Adaptive atmospheric modeling: key techniques in grid generation, data structures, and numerical operations with applications, Springer Verlag, 2006.

[7] Block, B. J., Lukáčová-Medviďová, M., Virnau, P., Yelash, L. Accelerated GPU Simulation of Compressible Flow by the Discontinuous Evolution Galerkin Method, *Eur. Phys. J. ST* **210** (2012), 119-132.

[8] Carr, L.E. III, Borges, C.F., Giraldo, F.X. An element-based spectrally-optimized approximate inverse preconditioner for the Euler equations, *SIAM J. Sci. Comp.*, **34** (2012), B392-B420 .

[9] Cockburn, B., Shu, C.W. The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM *J. Numer. Anal.* **35** (1998), 2440-2463.

[10] Cordier, F., Degond, P., Kumbaro, A. An asymptotic-preserving all-speed scheme for the Euler and Navier-Stokes equations. *J. Comput. Phys.* **231**(17) (2012), 5685-5704.

[11] Dolejší, V., Feistauer, M. A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *J. Comput. Phys.* **198** (2004), 727-746.

[12] Dudzinski, M., Lukáčová-Medviďová, M. Well-balanced bicharacteristic-based scheme for multilayer shallow water flows including wet/dry fronts, *J. Comput. Phys.* **235** (2013), 82-113.

[13] Feistauer, M., Dolejší, V., Kučera, V. Numerical simulation of compressible flow with a wide range of the Mach number by the discontinuous Galerkin method. *IASME Transactions* **6** 2 (2005), 964-973.

[14] Giraldo, F.X., Restelli, M. High-order semi-implicit time-integrators for a triangular discontinuous Galerkin oceanic shallow water model, *Int. J. Numer. Meth. Fl.* **63** (2010), 1077-1102.

[15] Giraldo, F.X., Restelli, M. A Study of Spectral Element and Discontinuous Galerkin Methods for the Navier-Stokes Equations in Nonhydrostatic Mesoscale Atmospheric Modeling: Equation Sets and Test Cases, *J. Comput. Phys.* **227** (2008), 3849-3877.

[16] Giraldo, F.X, Restelli, M., Lauter, M. Semi-implicit formulations of the Navier-Stokes equations: application to nonhydrostatic atmospheric modeling, *SIAM J. Sci. Comput.* **32** (2010), 3394-3425.

[17] Giraldo, F.X, Wartburton, T. A high-order triangular discontinuous Galerkin oceanic shallow water model, *Int. J. Numer. Meth. Fl.* **56** (2008), 899-925.

[18] Grabowski, W.W., Clark, T.L. Cloud–Environment Interface Instability: Rising Thermal Calculations in Two Spatial Dimensions, *J. Atmos. Sci.* **48**, (1991), 527-546.

[19] Harten, A., Lax, P.D., van Leer, B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, *SIAM Review* **25**(1), (1983), 35-61.

[20] Hartmann R., Houston, P. Symmetric interior penalty DG methods for the compressible Navier-Stokes equations. I., Method formulation, *Int. J. Numer. Anal. Model.* **3** (2006), 1-20.

[21] Heus, T., Jonker, H.J.J. Subsiding shells around shallow cumulus clouds, *J. Atmos. Sci.* **65**, (2008), 1003-1018.

[22] Hundertmark, A., Lukáčová-Medviďová, M., Prill, F. Large Time Step Finite Volume Evolution Galerkin Methods, *J. Sci. Comp.* **48** (2011), 227-240.

[23] Klein, R. Semi-implicit extension of a Godunov-type scheme based on low Mach number asymptotics. I. One-dimensional flow. *J. Comput. Phys.*, **121**(2) (1995), 213-237.

[24] Klein, R., Botta, N., Schneider, T., Munz, C.-D., Roller, S., Meister, A., Hoffmann, L., Sonar, T. Asymptotic adaptive methods for multi-scale problems in fluid mechanics. *J. Engrg. Math.* **39**(1-4) (2001), 261-343. Special issue on practical asymptotics.

[25] Klemp, J.B., Skamarock, W.C., Dudhia, J. Conservative split-explicit time integration methods for the compressible nonhydrostatic equations, *Monthly Weather Rev.* **135** (2007), 2897-2913.

[26] Klemp, J.B., Wilhelmson, R.B. Simulation of 3-dimensional convective storm dynamics, *J. Atmos. Sci.* **35** (1978), 1070-1096.

[27] Lukáčová-Medviďová, M., Noelle, S., Kraft, M. Well-balanced finite volume evolution Galerkin methods for the shallow water equations, *J. Comp. Phys.* **221** (2007), 122-147.

[28] Lukáčová-Medviďová, M., Morton, K.W. Finite volume evolution Galerkin methods: A survey, *Indian J. Pure & Appl. Math.* **41**(2) (2010), 329-361.

[29] Lukáčová-Medviďová, M., Morton, K. W., Warnecke, G. Evolution Galerkin methods for hyperbolic systems in two space dimensions, *MathComp.* **69** (2000), 1355-1384.

[30] Lukáčová-Medviďová, M., Morton, K. W., Warnecke, G. Finite volume evolution Galerkin methods for hyperbolic systems, *SIAM J. Sci. Comp.* **26(1)** (2004), 1-30.

[31] Lukáčová-Medviďová, M., Saibertová, J., Warnecke, G. Finite volume evolution Galerkin methods for nonlinear hyperbolic systems, *J. Comp. Phys.* **183** (2002), 533-562.

[32] Lukáčová-Medviďová, M., Warnecke, G., Zahaykah, Y. Finite volume evolution Galerkin (FVEG) methods for three-dimensional wave equation system, *Appl. Num. Math.* **57**(9) (2007), 1050-1064.

[33] Müller, A., Behrens, J., Giraldo, F.X., Wirth, V. An adaptive discontinuous Galerkin method for modeling cumulus clouds, Proceedings of the V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010, (2010), 1-20.

[34] Müller, A., Behrens, J., Giraldo, F.X., Wirth, V. Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2D bubble experiments *J. Comput. Phys.*, **235** (2013), 371-393.

[35] Park, J.-H., Munz, C.-D. Multiple pressure variables methods for fluid flow at all Mach numbers. *Internat. J. Numer. Methods Fluids* **49**(8) (2005), 905-931.

[36] Prill, F., Lukáčová-Medviďová, M., Hartmann, R. Smoothed aggregation multigrid for the discontinuous Galerkin method, *SIAM J. Sci. Comput.* **31** 5, (2009), 3503-3528.

[37] Robert, A. Bubble convection experiments with a semi-implicit formulation of the Euler equations, *J. Atmos. Sci.* **50**, (1993), 1865-1873.

[38] Restelli, M. Semi-Lagrangian and semi-implicit discontinuous Galerkin methods for atmospheric modeling applications, Ph.D. thesis, Politecnico di Milano, 2007.

[39] Restelli, M., Giraldo, F.X. A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling, *SIAM J. Sci. Comput.* **31** (2009), 2231-2257.

[40] Riviere, B. Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation, SIAM (2008).

[41] Schattler, U., Doms, G., Steppeler, J. Requirements and problems in parallel model development at DWD, Scientific Programming **8** (2000), 13-22.

[42] Skamarock, W., Oliger, J., Street, R.L. Adaptive grid refinement for numerical weather prediction, *J. Comput. Phys.* **80** (1989), 27-60.

[43] Skamarock, W.C., Klemp, J.B. Efficiency and accuracy of the Klemp–Wilhelmson time-splitting technique, *Monthly Weather Review* **122**, (1994), 2623-2630.

[44] Smolarkiewicz, P.K., Grabowski, W.W. The multidimensional positive advection transport algorithm: nonoscillatory option, *J. Comput. Phys.* **86** (1990), 355-375.

[45] Smolarkiewicz, P.K., Pudykiewicz, J.A. A class of semi-Lagrangian approximations for fluids, *J. Atmos. Sci.***49** (1992), 2082-2096.

[46] Straka, J.M., Wilhelmson, R.B., Wicker, L.J., Anderson, J.R., Droegemeier, K.K. Numerical solutions of a non-linear density current: A benchmark solution and comparisons, *Int. J. Numer. Fl.* **17**, (1993), 1-22.

[47] Thomas, S., Girard, C., Doms, G., Schattler, U. Semi-implicit scheme for the DWD Lokal-Modell, *Meteorol. Atmos. Phys.* **73** (2000), 105-125.

[48] Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics (A practical introduction)*, Springer 1999.

[49] Xue, M., Droegemeier, K.K., Wong, V. The advanced regional prediction system (ARPS) - A multiscale nonhydrostatic atmosphere simulation and prediction model. Part I: Model dynamics and verification, *Meteorology and Atmospheric Physics* **75** (2000), 161-193.